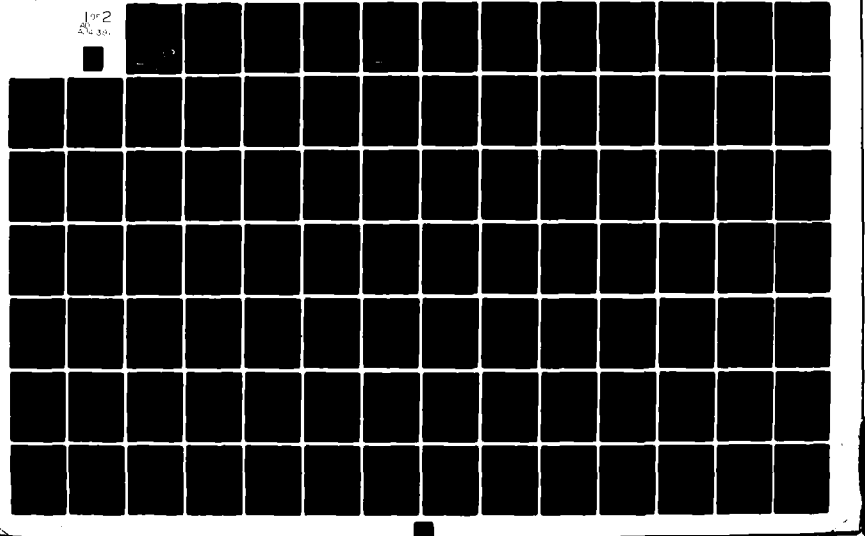
AD-A114 391 RAND CORP SANTA MONICA CA F/G 15/3
TSAR USER'S MANUAL. VOLUME III. VARIABLE AND ARRAY DEFINITIONS,--ETC(U)
FEB 82 D E EMERSON F49620-82-C-0018
UNCLASSIFIED RAND/N-1822-AF NL



(12)

# A RAND NOTE

TSAR USER'S MANUAL: VOLUME III--VARIABLE AND ARRAY DEFINITIONS, AND OTHER PROGRAM AIDS FOR THE USER

Donald E. Emerson

February 1982

N-1822-AF

Prepared For

The United States Air Force





Rand
SANTA MONICA, CA. 90406

The research reported here was sponsored by the Directorate of Operational Requirements, Deputy Chief of Staff/Research, Development, and Acquisition, Hq USAF, under Contract F49620-82-C-0018. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.



Published by The Rand Corporation

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>N-1822-AF | 2. GOVT ACCESSION NO.<br>AD-A114391 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>TSAR User's Manual: Volume III - Variable and Array Definitions, and Other Program Aids for the User | | 5. TYPE OF REPORT & PERIOD COVERED<br>Interim<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Donald E. Emerson | | 8. CONTRACT OR GRANT NUMBER(s)<br>F49620-82-C-0018 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>The Rand Corporation<br>1700 Main Street<br>Santa Monica, CA. 90406 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Requirements, Programs & Studies Group (AF/RDQM)<br>Ofc, DCS/R&D and Acquisition<br>Hq USAF, Washington, DC 20330 | | 12. REPORT DATE<br>February 1982<br>13. NUMBER OF PAGES<br>125 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>Unclassified<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report)<br>Approved for Public Release; Distribution Unlimited | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)<br>No Restrictions | | |
| 18. SUPPLEMENTARY NOTES | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number)<br>Military Air Facilities<br>Logistics<br>Attacks<br>Computer Programs<br>Computerized Simulation | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number)<br>See Reverse Side | | |



DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

This Note is one of five documents that collectively describe the TSAR and TSARINA computer models developed to assess the effect of air attacks on the sortie generation capabilities of air bases. The Theater Simulation of Airbase Resources (TSAR) model provides an analytic context within which a variety of airbase improvements may be tested. The present volume of the User's Manual will be useful primarily for those that are interested in modifying and extending the existing program logic, or are trying to understand apparent errors.

(12)

# A RAND NOTE

TSAR USER'S MANUAL: VOLUME III--VARIABLE AND ARRAY DEFINITIONS, AND OTHER PROGRAM AIDS FOR THE USER

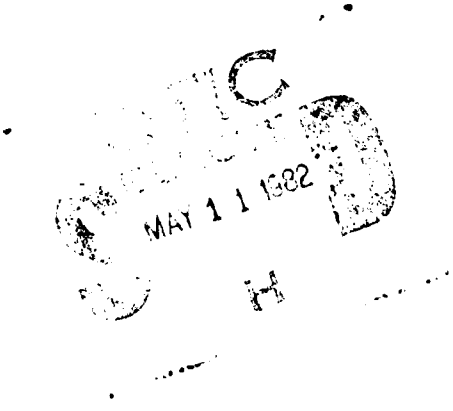
Donald E. Emerson

February 1982

N-1822-AF

Prepared For

The United States Air Force




Rand
SANTA MONICA, CA. 90406

# PREFACE

This Note is one of five documents that collectively describe the TSAR and TSARINA computer models developed at The Rand Corporation to assess the effect of air attacks on the sortie generation capabilities of air bases. This development was carried out under the Project AIR FORCE Resource Management Program project entitled, "Strategies To Improve Sortie Production in a Dynamic Wartime Environment."

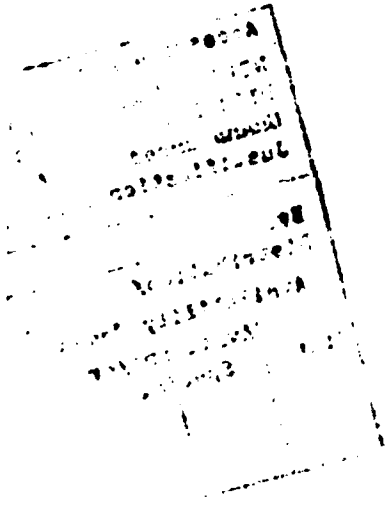
The Theater Simulation of Airbase Resources (TSAR) model provides an analytic context within which a variety of airbase improvements may be tested. New passive defenses, new maintenance doctrine, modified manning levels, improved base repair and recovery capabilities, increased stock levels for parts and equipment, etc., as well as concepts for improved theater-wide resource management, all can be examined for their effect on aircraft sortie generation. These models have been briefed to several Air Force organizations during the development process.

This volume of the User's Manual will be useful primarily for those that are interested in modifying and extending the existing program logic, or are trying to understand apparent errors. The companion documents include:

| | |
|---|---|
| R-2584-AF | Introduction to the TSAR Simulation Program Model Features and Logic |
| N-1460-AF | TSARINA--User's Guide to a Computer Model for Damage Assessment of Complex Airbase Targets |
| N-1820-AF | TSAR User's Manual: Vol. I--Program Features, Logic, and Interactions |

N-1821-AF TSAR User's Manual: Vol. II--Data Input, Program Operation and Redimensioning, and Sample Problem

Other documents are planned that will discuss the problems associated with data acquisition for these models, and will present the procedures currently under development at Rand to assist in solving those problems.



## ACKNOWLEDGMENTS

The development of the TSAR computer model demanded uninterrupted concentration over an extended period. My debts to the Air Force and to Rand management are obvious. Not so obvious are the debts owed my most understanding family, who endured my total absorption in TSAR's development for over three years.

Among my colleagues at Rand, I would particularly like to thank Louis Wegner and Michael Poindexter for their many helpful ideas and suggestions for dealing with a variety of programming problems, and Major John Halliday and Milton Kamins for their ideas that have been incorporated into TSAR logic and for their careful work in creating the data bases that were used for TSAR development.

CONTENTS

## GLOSSARY

| | |
|---|---|
| AGE | Aerospace ground equipment and other equipment used for carrying out various tasks |
| AIS | Avionics Intermediate Shops; special test equipment used for repairing avionic LRUs and SRUs. |
| BLSS | Base-level self sufficiency stock of aircraft spare parts |
| CAP | Combat Air Patrol |
| CAS | Close Air Support |
| CILC | Centralized Intermediate Logistics Concept |
| CIRF | Centralized Intermediate Repair Facility |
| COB | Collocated Operating Base |
| COMO | Combat Oriented Maintenance Organization |
| CONUS | Continental United States |
| FRAG | Fragmentary order that specifies flight requirements |
| LCOM | Logistics Composite Model |
| LRU | Line replaceable unit; an aircraft spare part |
| MOB | Main Operating Base |
| NMCS | Not mission capable because of lack of spare parts |
| NORS | Not operationally ready because of lack of spare parts; same as NMCS |
| NRTS | Not reparable this station |
| OST | Order and ship time in days |
| PAA | Program Authorization, Aircraft |
| POL | Petroleum, oils and lubricants; often used as an abbreviation for aircraft fuel |
| POS | Peacetime operating stock; an organization's stock of aircraft spare parts for aircraft maintenance in peacetime |

| | |
|---|---|
| RAM | Rapid area maintenance; special mobile teams used for repairing aircraft battle damage |
| RR | Flight line maintenance that removes and replaces malfunctioning aircraft parts with serviceable components |
| RRR | Flight line maintenace that removes, repairs, and replaces aircraft spare parts (actually, usually removes and replaces with a serviceable unit, and then repairs malfunctioning unit) |
| RRR | Rapid runway repair |
| SAMSOM | Support Availability Multi-System Operations Model |
| SCL | Standard combat load that designates the mission dependent munitions to be loaded |
| SRU | Shop replaceable unit; a component of an LRU |
| TRAP | Tanks, racks, adaptors and pylons |
| TSAR | Theater Simulation of Airbase Resources |
| TSARINA | TSAR Inputs using AIDA |
| WRM | War Reserve Material |
| WRSK | Wartime readiness spares kit |

# APPENDIX A

## VARIABLES IN COMMON

Definitions for most of the 270 variables carried in one or another of the several major blocks of Common data are listed in this section in alphabetical order. The remainder are listed below in the table of array dimensions and control data:

| ARRAY NAME | ARRAY LENGTH | CURRENT NUMBER OF ENTRIES | FIRST EMPTY LOCATION | LAST EMPTY LOCATION | OVERFLOW TALLY |
|---|---|---|---|---|---|
| ACN | MAXACN | NEH | --- | --- | OVERA |
| BACKLG | LLQ | NEL | FEL | LEL | OVERL |
| BUILDQ | LBQ | NEB | FEB | LEB | OVERB |
| CEJOBQ | LTHCEQ | NEC | FEC | LEC | OVERC |
| CHANGE | NOCHG | NEV | FEV | LEV | OVERV |
| DEFTSK | LDT | NED | FED | LED | OVERD |
| FLTRQT | LFQ | NEF | FEF | LEF | OVERF |
| INTTSK | LIQ | NEI | FEI | LEI | OVERI |
| LIMBO | NLIMBO | NEX | FEX | LEX | OVERX |
| NORQ | LNOR | NEO | FEO | LEO | OVERO |
| PILOT | NOCREW | NPILOT | --- | --- | OVERM |
| REPQ | LRQ | NER | FER | LER | OVERR |
| RESUPP | LGQ | NEG | FEG | LEG | OVERG |
| RQDTSK | LNT | NEN | FEN | LEN | OVERN |
| SHIP | NOSHIP | NES | FES | LES | OVERS |
| SHIPQ | NOPKG | NEP | FEP | LEP | OVERP |
| SHPTSK | NOTASK | --- | --- | --- | OVERK |
| TASKQ | LTQ | NET | FET | LET | OVERT |
| WAITSK | LWQ | NEW | FEW | LEW | OVERW |
| REJOIN | NJOINT | NEJ | FEJ | LEJ | OVERJ |

ADAPTR — NRTS policy for RR parts is changed when there are fewer LRUs than ADAPTR percent of initial LRU stocks; they are shipped to lateral resupply base, rather than nominal NRTS destination.

ALERTR — Is set to unity if personnel are required to be assigned for alert aircraft.

AIDA — Is zero unless the base damage input data are generated with the TSARINA model. When not zero the resource damage data may be specified both for specific types and for all other types. AIDA is set to two if the expected value mode is used in TSARINA for computing damage to aircraft shelters; otherwise it is one.

ATRISK — When a shop facility or all elements of a distributed shop are damaged at the time of a subsequent attack the resources assigned to that shop are assumed to have been relocated and to be invulnerable if ATRISK is zero; if ATRISK is unity, the damage is assessed as though operations were normal.

AVGTT — The average shipment time, in hours, from a CIRF to the operating bases; computed internally.

BUILD — If unity, the munitions buildup features are activated.

CANCAN — Is set to unity when a part may be cannibalized even though there is a reparable part on base.

CANFLT — A flag that is set to unity when the remaining segments of a composite flight must be canceled.

CANMOD — Cannibalization mode (see subroutine CANNIB).

CANMUL — Task time when a part is cannibalized; expressed as a percentage of the nominal time for the task segment that specifies the part (default = 150).

CANSRU — If greater than zero, the SRUs are removed from an LRU that is waiting for repair at an operating base, if aircraft are NORS because of the LRU; at a CIRF, an LRU is "cross-canned" if CANSRU aircraft in the theater have this LRU missing.

CCIRF — Control mode for CIRF operations.

CDELAY — The default time for cannibalization is one-half the related on-equipment task time, plus CDELAY minutes.

CEAGE — The number of types of equipment associated with civil engineering tasks.

CEDELY — Initiation of all reconstruction tasks is delayed by this number of minutes after an airbase attack, to account for the preliminary delays involved in overcoming the disruptive effects of fires, roadway damage, etc. This delay is modified to reflect the weight of the attack when TSARINA's variable delay option is used.

CEPEO — The number of types of personnel associated (exclusively) with civil engineering tasks.

CEWORK — Switch; when = 1, civil engineering resources are allocated to repair damage from airbase attacks in accord with the priorities defined by the CEPRTY array.

CIRFLG — Lateral resupply flag; is set to unity when part is to be taken from first base that can release a part; it is set to 2 if the base best able to provide the part is to be sought.

CONCUR — If unity, battle damage repair jobs may be initiated concurrently with the first of the other unscheduled maintenance tasks; otherwise, the battle damage tasks are scheduled before all other maintenance activity.

CONSIG — If zero, any parts that are shipped to the theater to replace condemned parts, and LRUs that were NRTSed to CONUS, are consigned to the base of origin on return; if unity, all parts are consigned to the theater manager for distribution.

CRASH — When runways are closed at all operating bases (and at any emergency base) recovering aircraft will be lost if this variable is initialized to unity; if not initialized, the sortie length is artifically extended such that the aircraft will land after the runway at the planned recovery base has been opened.

CRBLDG — Unless civil engineering resources are sufficient to initiate repairs to all damaged facilities up to and including this priority, reconstruction tasks are pursued with secondary procedures using lesser resources.

CREWS — Air crews are accounted for when = 1; neglected if 0.

CUMSTA — If 0 the task time and delay data are cumulated for each trial; if = 1, the data are cumulated over all trials.

CMODE — When not zero defines the mode of operation for theater resource management (see Section XI).

CTHEA — Control mode for theater resource management.

C4INT — Time interval in hours between periodic theater resource reviews, subsequent to the initial review.

C4TM — Time for initial theater resource review-hours.

DAMODE — Internally generated flag that denotes the mechanism being used to input damage data:

0 = User supplied card image damage data
1 = Single=attack damage stored on disk for all trials
2 = Separate damage data stored for each trial.

DELYPF — During the time that DELYPF is unity the preflight assignment task is delayed until LOADTM.

| | |
|---|---|
| DOLD | Number of the last aircraft to have a task entered into the deferred task array. |
| DOWNTM | Parts may not be cannibalized from an aircraft with a ready-to-fly time within "DOWNTM" hours. |
| EF | Pointer to the location of the earliest flight to be launched. |
| EMERG | Base designated for emergency recovery when runways are cut at all operating bases; may be the same as a rear maintenance base. (See p. 80, Vol. II, for unique task sequence requirements at this base.) |
| ENDAY | End of the nominal flying day (used to control accomplishment of deferred maintenance) - hours. |
| EXPED | When greater than zero, the parts repair and equipment repair administrative delays are reduced by 1/EXPED when there are no serviceables available. |
| EXTEND | When initialized to unity, an NTRIAL simulation is a one-trial simulation of NTRIAL × SIMLTH days. |
| FA | Pointer to the next arrival of an intra-theater shipment. |
| FD | Pointer to the next departure of an intra-theater shipment. |
| FILLAC | Controls use of filler force aircraft. |
| FIXAGE | Set to unity if equipment may be broken and repaired. |
| FLEVEL | When zero, augmentee and filler aircraft are managed so as to maintain the number of aircraft on base equal to the assigned numbers; when unity, the non-battle damaged aircraft are maintained equal to the assigned number; when two or three, the aircraft or aircraft without battle damage are maintained equal to the capacity of the available aircraft shelters. |
| FSALVG | If an aircraft is damaged by air attack and is not reparable, FSALVG percent of the aircraft's spare parts not destroyed during the attack are salvaged and added to the serviceable. |
| FULL | If unity, all parts are on base, none enroute, at time zero (identified as NOPIPE in Common). |
| HIATUS | Delivery of parts in the pipeline at the beginning of the simulation are delayed HIATUS days. |
| HPEO1<br>HPEO2 | Used for transferring the identity of personnel scheduled for release among subroutines. |

| | |
|---|---|
| IDAY | Number of days for which the sortie data are aggregated across trials; IDAY is 1 if SIMLTH is 30 or fewer days, 2 if from 31 to 60, and 3 if between 61 and 66 days. |
| IGNORE | When initialized to unity, all jobs that may be deferred for all missions are ignored. |
| INDEX | A threshold used when checking repair jobs waiting at a CIRF; if exceeded as jobs are checked, the job is processed without checking for a higher priority job. The appropriate value to set will depend upon which of the two logics (PRTY1 or PRTY3) is in use. |
| ISHORT | Parts shortfall from "authorized" levels (percent). |
| ITRIAL | Number of the current trial. |
| JOBCON | Controls extent of rear-base maintenance. |
| K1LOW<br>K2LOW | For parts that are "critically short" (see TOOFEW), the actual stock level, as a percentage of the nominal stock level, is selected at random in the range K1LOW and (K1LOW + K2LOW). |
| LA | Pointer to the last scheduled arrival of an intra-theater shipment. |
| LD | Pointer to the last scheduled departure of an intra-theater shipment. |
| LF | Pointer to the location of the last flight to be launched. |
| LOADTM | Nominal time of day to commence preflight preparation for the day (hour). |
| LOSTAC | Cumulative number of aircraft lost in air operations and air base attacks. |
| LSTTOD | Last time of day for commencing morning preflight (used to limit expected time for deferred tasks) (an even-numbered hour only). |
| LTHATT | Length of ATTACK array; maximum number of airbase attacks that may be scheduled during a simulation. |
| LTHCEQ | Length of CEJOBQ array; maximum number of simultaneous civil engineering tasks. |
| LTHLST | Length of LISTIN array; maximum number of task-incompatibility descriptors. |

| | |
|---|---|
| MULTI | Number of flights stored temporarily while checking for a composite flight. |
| MAXACN | Maximum number of aircraft that can be accommodated (size of the ACN array). |
| MAXB | Maximum number of airbases (limit = 9). |
| MAXFLT | Current number of periodic flight schedules entered in the PRDFLT array. |
| MAXM | Maximum number of missions for each type of aircraft (limit = 5). |
| MAXMNT | When the projected maintenance time exceeds MAXMNT, a filler aircraft may be requistioned (see also FILLAC and FLEVEL). |
| MAXPER | Maximum number of periodic flight schedules that may be stored; dimension of the PRDFLT array. |
| MAXREC | Maximum number of items stored in daily aircraft activity report; dimension of RECORD array. |
| MAXT | Maximum number of aircraft types (limit = 9). |
| MAXTME | The time remaining for deferred maintenance before LSTTOD (reassessed every 30 minutes in MANAGE). |
| MNTLMT | If the time estimated for getting an aircraft ready to fly exceeds MNTLMT (and certain other conditions are fulfilled--see Card Type #3/2), the aircraft is ferried to a rear base for the required maintenance. |
| MNTF<br>MNTR | In considering whether an aircraft is to be sent to the rear for maintenance, MNTF and MNTR are used in the decision algorithm (see Card Type #3/2). |
| MULTI1 | When a base's projected sortie generation capability per assigned aircraft is greater by MULTI1 percent than that of the parent base of an aircraft, the aircraft is retained and is not returned to the parent base. |
| MULTI2 | Aircraft reassignment (effective if STATE = 3) occurs among bases whose projected sorties per available aircraft differ by more than MULTI2 percent. |
| MXHOLE | The maximum number of missing parts (holes) that are permitted on a particular aircraft (default = 10000). |
| NBASE | The number of bases that operate aircraft. |

| | |
|---|---|
| NCARGO | Length of the CARGO array; maximum number of items in the support shipments from CONUS. |
| NCONUS | Number of the next shipment from CONUS. |
| NESHP | Number of daily intra-theater shipments. |
| NEWDTA | The time at which theater resource reports are to be initiated; only applicable if OLDATA is initialized as zero. |
| NEWPRT | A switch: when unity, the automatic parts initialization computations are repeated for each trial. |
| NEXTIN | Next time that the sortie demand data are to be read and reorganized. |
| NEXTSC | Next time that the sortie demand data are to be reorganized. |
| NJOINT | Length of the REJOIN array. |
| NLIMBO | Length of the LIMBO array; maximum number of parts and equipments undergoing an administrative delay. |
| NOAGE | Length of AGESTK array; maximum number of types of AGE and other equipment. |
| NOATT | Number of attacks stored in the ATTACK array. |
| NOBILD | Length of the MUNRQT array; maximum number of munition assembly procedures. |
| NOCE | Length of CERQTS array; maximum number of civil engineering procedures. |
| NOCHG | Length of the CHANGE array; maximum number of scheduled parameter changes. |
| NOCREW | Length of the PILOT array; maximum number of air crews accommodated in the theater. |
| NOCONF | Length of the CONFIG array; maximum number of configurations. |
| NOCONS | Length of CONUS array; maximum number of support shipments from CONUS. |
| NOFAC | Length of FACLTY array; maximum number of airbase facilities. |
| NOFUEL | If unity, other preflight tasks are prohibited during refueling. |

NOITEM Length of DAMAGE array; maximum number of damage data for airbase attacks.

NOLD Number of the last aircraft to have a task stored in the required task array (RQDTSK).

NOMATL Length of the MATERL array; maximum number of types of materials for civil engineering tasks.

NOMUN Length of MUNSTK array; maximum number of types of munitions.

NONUNI Switch; when unity, resource losses determined by sample from binomial distribution.

NOPART Length of PARTS array; maximum number of parts that may be specified.

NOPEOP Length of PEOPLE array; maximum number of personnel types that may be specified.

NOPIPE When unity, all parts computed by the automatic parts generation logic are on-base at the beginning of the simulation; none are in the parts pipelines.

NOPKG Length of SHIPQ array; maximum number of sets of items that may be waiting for intra-theater shipment.

NOPOMO The average additional on-equipment task time that is required at a base operating under 66-1, when the data apply to 66-5 activities.

NOPRT Length of the PRTCRT and PRTLST arrays (need only equal the highest position in the PARTS array in which a part or LRU is found).

NOREP Length of REPRQT array; maximum number of parts repair procedures.

NOREPA Length of REPALT array; maximum number of alternate parts repair procedures.

NOREPT Length of REPORT array; maximum number of resource reports that may be scheduled during a single day.

NORPT Number of report times stored in the REPORT array.

NOSAVE When NOSAVE = 1, records are not saved for parts that break after an air attack has closed the shop that would normally process the repair, if the projected shop reconstitution time is not earlier than the end of the simulation.

| | |
|---|---|
| NOSCL | Length of SCLRQT array; maximum number of combat loadings that may be specified. |
| NOSHIP | Length of SHIP array; maximum number of intra-theater shipments that may be scheduled at one time. |
| NOSHP | Length of SHIPSC array; maximum number of daily intra-theater shipments that may be stored. |
| NOSTAT | Maximum number of types of AIS stations in the simulation. |
| NOTASK | Maximum number of tasks in each shop group for each type of aircraft (must be a multiple of 4). |
| NTRIAL | Number of repetitions of the simulation. |
| NOTRAP | Length of TRAP array; maximum number of TRAP types. |
| NOTRAY | Maximum number of trays (one for each LRU) in the simulation. |
| NOTSK | Length of TSKRQT array; maximum number of on-equipment maintenance procedures. |
| NOTSKA | Length of TSKALT array; maximum number of alternative on-equipment task procedures. |
| NOUSER | Length of the BORROW array. |
| NOW | Current simulation time (TTU). |
| NPART | The number of the highest numbered LRU or SRU (default is NOPART). |
| NRTPOL | If unity, an LRU that requires an unavailable SRU that is not nominally stocked is NRTSed. |
| NSCROL | Maximum number of aircraft whose activities may be stored in the RECORD array (maximum = 24). |
| NTYPE | Number of aircraft types to be employed in the simulation (may be less than, or as great as, MAXT). |
| NXSCH | Next time at which the intra-theater shipments are to be rescheduled. |
| OLDATA | Base resource reports are generated when zero and deferred initially while equal to unity. |

OPSBSE — Number of bases used in the simulation for supporting combat operations; excludes rear maintenance bases and the emergency recovery base, when one is used.

ORDER1 — Threshold controlling CIRF response to parts shortages; responds only if (Enroute Parts + On-base Repairables - Required Parts) is less than or equal to ORDER1.

ORDER2 — Threshold controlling an operating base's recourse to lateral resupply; seeks lateral resupply only if (On-base Reparables - Required Parts) is less than or equal to ORDER2. (Reparables are assessed only if the shop is open and functioning.)

ORDIT — Interrupted tasks and repairs are prioritized when ORDIT = 1; FIFO if 0.

ORDWT — Waiting tasks and repairs are prioritized when ORDWT=1; FIFO if 0.

OUTFIT — Activates the automatic parts stock initialization.

OVERFL — Value controls simulation behavior if the dimensions of the arrays used to store internally generated data are exceeded:

When OVERFL = 0, simulation stops;
= 1, overflow noted and tabulated;
= 2, overflow noted for first entry and tabulated;
= 3, overflow tabulated.

This feature must be used with caution because program behavior can become highly erratic when records are discarded. In any event execution is terminated automatically at the end of any day when the cumulative number of discarded records is 20 or more.

OVERTM — Number of minutes of overtime permitted.

PKGTM — Number of minutes required to package resources for intra-theater shipment.

PMODE — When unity, parts initialization for WRSKs approximate DO-29; otherwise the Chapter 11 procedures from AFM 67-1 apply.

POSTPN — If zero, all unscheduled maintenance tasks must be accomplished prior to next flight; if = 1, tasks will be deferred (postponed) that are not critical for next mission.

PPRINT — Controls output summaries of the initial stock levels and the parts pipelines (see subroutine IPARTS). When increased by 10, residual parts levels are listed after the delay statistics.

PRINT — Value controls content of simulation output.

PROTME — When insufficient aircraft are ready for a scheduled flight, and none can be found in the spare queue or a lower piority alert, an aircraft can be taken from another scheduled flight of the same or lower priority if the flight time is at least PROTME minutes later (default = 30 minutes).

QUIK — If zero, filler aircraft are launched when the aircraft being replaced are sent to the rear; if unity, the filler aircraft are launched as soon as the decision is made to send aircraft to the rear.

RANDM — When unity, parts shortages and the location of parts in the pipelines are determined with samples from the Poisson approximation of a binomial distribution.

REST — Minimum number of minutes for aircrews between flights.

RPARTS — User specified fraction of the generated spare parts stocks that are placed at the rear maintenance base to service aircraft taken to the rear because the estimated ready-to-fly time exceeded MNTLMT.

SCROLL — When unity, daily activity reports are preserved for 24 specified aircraft for a specified number of days.

SCROL1 — Aircraft number of the first of the 24 aircraft for which a daily activity report is prepared.

SCROL2 — Number of last of 24 aircraft for which a daily activity report is prepared.

SEED — If not zero, the value used for the seed of the random-number generator is controlled (i.e., is prespecified by the local operating system).

SEEKSH — When unity, another in-theater shop is sought for parts repair when the nominal shop is closed by damage.

SELECT — When not zero, a daily summary of sortie demands is prepared to facilitate selection of bases for sorties.

- $\geq 1$ — Summary data used when base not specified
- $\geq 2$ — Summary data used for reallocating demands on airbases with closed runways

SHOPRY — Controls the choice of rules for prioritizing repairs at a CIRF.

SHPDLY — This delay is introduced to all on- and off-equipment aircraft-related tasks, to account for the disruption following an airbase attack. It is modified to reflect the weight of the attack when TSARINA's variable delay option is used.

SHPREP — If not zero, all parts repaired at an operating base are shipped to the base selected with the SEND logic in the CONTRL subroutine, when (On-base NORS Aircraft - Required Parts) is greater than, or equal to, SHPREP.

SIMLTH — Length in days of the period to be simulated.

SLEEP — Minimum number of off-duty hours between shifts.

SPARE1 ... SPARE9 — Nine undefined variables included in the BASIC common for future requirements.

STATE — If not zero, the state of each base's capability to generate sorties is computed daily.

- $\geq 1$ Base-state-data used to select base for diversion
- $\geq 2$ Base-state-data used to decide when aircraft recover at their parent base (see MULTI1)
- $\geq 3$ Aircraft base assignment reorganized nightly when work-loads are disproportionate (see MULTI2)

STATFQ — The frequency, in days, with which the summary data regarding the average length of time for tasks and jobs, and the causes and lengths of the aircraft delays, are printed. If STATFQ = 0, these data are not collected or printed.

TCONUS — Time of the next shipment from CONUS.

TEST — Controls internal debugging features. If >0 diagnostic messages are printed for the entire simulation; if -1, a special card must follow that defines the number of the trial and the time intervals during which the debugging data are required.

TEST1 — The value for TEST during the specified intervals for debugging output.

TODOCK — If unity, parts that are normally NRTSed to another base but can't be because no shipment schedule exists, are held for later lateral repair rather than being sent to CONUS.

TOOFEW — If positive, the parts supply system is critically short of a percentage of aircraft spare parts (equal to TOOFEW/10); the part numbers that are short are selected at random. If "-1", the probability a part is short is proportional to the cost of the part.

TPLAN — Time that aircraft supply and demand were last projected.

TSAR — Controls theater resource management; is initialized to unity if the management of resources is to be centralized; is initialized to "2" if the theater also has a CIRF for parts repair.

TTRIAL — The number of the trial during which special debugging data is to be output.

UNCER — When initialized with the number of a distribution from the TTIME subroutine, the "actual" unscheduled maintenance task probabilities used in the simulation are determined by selecting a value from that distribution, assuming the mean is the value entered by Card Type #7. (Parts initialization and shop activity at zero time are based on the values entered--i.e., "peacetime" data points.)

VBREAK — A switch. If zero or -1, unscheduled maintenance task probabilities are modified in proportion to the Card Type #44 entries. If unity the basic probabilities are varied by shop and aircraft type as a function of achieved sortie rate. If set to -1 or +1, the basic values are used for estimating average shop task times, average resource requirements (in BSECAP) and initial parts stocks.

VERIFY — If set to 1, 2, or 3, additional checks are conducted of input data accuracy. When set to 2 or 3, either by the user, or automatically when certain input errors are detected, execution is stopped following data entry. When = 3, most card types are subjected to special checks in subroutine TESTER.

WHEN2 — Used for transferring the initiation time of a task element among certain subroutines.

WXDAYS — Maximum of days for which weather data may be stored. (maximum value = 65)

XTEST — If initialized when VERIFY = 2, TEST is set to XTEST for the last portion of the initialization process carried out by subroutine TRIALS.

ZNORS — A switch. When unity, parts that were not available to be placed in the pipeline during parts initialization, because of shortages are obtained by removing them from aircraft, thus creating NMCS (NORS) aircraft. If zero, a message is printed noting the shortage.

ZSHOP — Internally set to unity when Card Type #42 is used to initialize on-equipment or off-equipment work at time zero.

## APPENDIX B

## DATA STORAGE ARRAYS IN COMMON

The 180 storage arrays used in TSAR and contained in one or another of the 20 Common statements (other than the LOCALx Common statements) are listed alphabetically in this section.

Data that are input primarily by the user are denoted by INPUT-#xx after the array descriptor; "xx" provides a cross-reference to the Card Type used for data input. Data that are generated internally are denoted by GENERATED.

The array name and dimensions follow an English-language descriptor. Dimensions controlled by the user are listed in terms of the variable that defines the particular dimension (see Appendix A). MAXB, MAXT and MAXM refer to the maximum numbers of bases, aircraft types and missions, respectively. Unless otherwise specified, the dimension of SHOP is 30. The nature of the stored data is then defined for each element along the program-fixed dimension.

In many cases more than one data are contained in a single element. The packing factor is shown as a multiplier using the FORTRAN symbol for multiplication; the first item is multiplied by the packing factor and then added to a second before storage; e.g., "Time*10/distribution" defines ten times a time plus a distribution number. As will be noted, the organization and packing of the data is often different than specified in the Card-Image input formats.

Aircraft Assigned (GENERATED)

ACA (I, MAXM, MAXT, MAXB)

I = 1 Number of aircraft assigned to 3rd priority flights
= 2 Number of aircraft assigned to 5th priority flights
= 3 Number of aircraft assigned to 6th priority flights

Aircraft Type Data (INPUT-#15)

ACDATA (I,MAXT)

I = 1 Post-flight inspection delay - Mean time*10/distribution
= 2 Pre-flight inspection delay - Mean time*10/distribution
= 3 Fuel - Thousands of pounds
= 4 Task number for fueling resources
= 5 Number of assignable mission types
= 6 Nominal time for unscheduled maintenance
= 7 Nominal time for complete sortie cycle
= 8 Pointer to lst item in PRTLST
= 9 Munitions load team personnel: Type*100/Number
(Enter to prohibit two teams per aircraft)
= 10 Special AGE#1: one unit is sufficient for all tasks
= 11 Special AGE#2: one unit is sufficient for all tasks
= 12 Basic munitions #1; Type*100/number
= 13 Basic munitions #2; Type*100/number
= 14 Basic munitions #3; Type*100/number
= 15 Administrative delay for transferred aircraft
= 16 First battle-damage task
= 17 Last battle-damage task
= 18 Percentage of parts that are recoverable from a salvaged aircraft
= 19 First airbase-attack-damage task
= 20 Last airbase-attack-damage task
= 21 Personnel required for an alert aircraft - Type*100/number
= 22 Equipment required for an alert aircraft - AGE1*100/AGE2
= 23 Number of sorties per PAA for which battle damage spares are to be procured
= 24 Base number where rear-base maintenance is performed
= 25 Unity if aircraft may be assigned to "special" alert
= 26-28 Mission effectiveness degradation when the (I-25)th basic munition is not loaded
= 29-30 Spare

Aircraft Mission Data (INPUT-#16)

ACMDTA (I, MAXM, MAXT)

I = 1 Flight duration - Mean time*10/distribution
= 2 Non-reparable damaged aircraft fraction*128/Damage to kill ratio*10
= 3 Late takeoff time allowance
= 4 Percent aborts per sortie*128/Percent crews lost when aircraft lost in combat
= 5 Equivalent percent sorties retaining mission-dependent munitions*128/Percent sorties retaining basic-munitions
= 6 Last day initial attrition rate*500/percent attrition*10 (maximum attrition rate is 49.9 percent per sortie)
= 7 Last day second attrition rate*500/second percent attrition*10
= 8-10 Third through fifth attrition rate data
= 11 If unity, aircraft in a flight land at the same time
= 12 Total theater sortie demand outstanding at beginning of current 2-hour period
= 13-20 Spare

Individual Aircraft Data Array (GENERATED)

ACN (MAXACN, I)

I = 1 Aircraft type *512/ Prior mission *64/ Parent base
= 2 Next base*100/Present base
= 3 Pointers: First interrupted task
= 4 Last " "
= 5 First waiting task
= 6 Last " "
= 7 First required task
= 8 First deferred task
= 9 First task in TASKQ
= 10 Next aircraft, same base
= 11 Next aircraft assigned to the same flight, same alert force, or same spare force
= 12 Status (1) In flight (2) PSTFLT delay (3) Maintenance (unassigned) (4) PREFLT delay (5) Maintenance (assigned) (6) Ready to fly (7) Deferred maintenance
= 13 Configuration*128/Standard combat load (SCL)
= 14 Earliest projected flight time (zero if ready to fly)
= 15 Designated mission*1000/Assigned FLT
= 16 Preflight flag
= 17 Current criticality based on deferred tasks; minus while the aircraft is waiting for a late launch
= 18 Time present delay is complete
= 19 Pointers: To delay time heap
= 20 Heap pointer
= 21 Actual completion time of longest in-process task
= 22 Number of "holes" in aircraft
= 23 Crew number for current flight
= 24 Squadron number when COMO maintenance is used
= 25 Number of tasks requiring munitions load crew; + 100 if crew at work; + 200 if crew held momentarily
= 26 Number of ongoing tasks that demand a unit of special AGE#1 be present
= 27 Number of tasks that demand a unit of AGE#2
= 28 Assignment status: 0 = Unassigned; 1 = Flight; 2 = Alert; 3 = Unassigned (spare) queue
= 29 Aircraft "hole" criticality (generated in ACCRIT)
= 30 Completion time of most recent sortie
= 31 Temporary flag used in FLIGHT and LAUNCH
= 32 Sorties initiated during the current day
= 33 Flag denoting status of rear-base maintenance
= 34 Sorties flown since a temporarily deferrable task arose
= 35 Sum of "1", if battle damage tasks are scheduled at operational base, and "10" if damage tasks are scheduled for a rear base
= 36 Pointer to a list of ongoing tasks that rejoin
= 37 Defines regular base when diverted for emergency recovery
= 38 Mission effectiveness degradation for omitted basic munitions*10/coded record of munitions that are loaded.
= 39-40 Spare

Administrative Time Delays for Parts and Equipment Repair(INPUT-#47)

ADELAY (SHOP(24), I, MAXB)

I = 1 When a faulty part is removed from an aircraft and sent to a shop for repair; or when a part arrives at a CIRF; the repair process is delayed this length of time, except when EXPED is not 0.
= 2 When a piece of equipment is found to be faulty, the repair process is delayed this length of time, except when EXPED is not 0.
Entry: Hours*100/Time distribution

Equipment Repair Procedures (INPUT-#10)

AGEREP (NOAGER, I)

I = 1 Shop assigned to repair AGE; or pointer to subsequent procedure
= 2 Probability AGE requires repair following use *10000; or, minus probability this particular repair procedure is required*100
= 3 Repair time Mean *10/Distribution;
or, minus AIS station number
= 4 Personnel Type *100/Number;
or, -1 if multiple procedures are to be considered
= 5 Equipment AGE1 *100/AGE2;
or, first procedure to be considered
= 6 Alternate procedure

Resource Report on AGE and Equipment (GENERATED)

AGERPT (NOAGE, MAXB)

Total number on base - Data received*128/Data in transit

AGE Requirements per Sortie (GENERATED)

AGERQT (NOAGE, MAXM, MAXT)

Likelihood needed*(10**7)/Expected requirements for AGE per sortie - (100000*TTU)

AGE and Equipment Stocks (INPUT-#22)

AGESTK (NOAGE, I, MAXB)

I = 1 Number serviceables on base*100/Nominal stock level
= 2 Number available*100/Nominal shop
= 3 Number serviceables enroute

Alert Aircraft Resources Flag (GENERATED)

AIDALT (MAXT)

Switch; set to unity if resources are required for alert aircraft

Data on AIS Station Equipment (INPUT-#22/66)

AISDTA (NOSTAT, I)

I = 1 Pointer to the first tray in TRAYS associated with the station
= 2 Part breakage probability per LRU repair
= 3 Part OST (days) *10/Distribution
= 4 Added time needed for AIS maintenance; a percentage of LRU repair time--with one station*128/with more stations
= 5 Equivalent AGE number of AIS station

Tray Utilization of AIS (GENERATED)

AISUSE (NOSTAT, I, MAXB)

I = 1 TRAY in use on string #1
= 2 TRAY in use on string #2
= 3 TRAY in use on string #3
= 4 TRAY in use on string #4
= 5 TRAY in use on string #5
= 6 Cumulative LRUs and SRUs repaired by this station type.

Alert Aircraft (GENERATED)

ALERT (I, MAXM, MAXT, MAXB)

I = 1 Number aircraft required for Priority #2 alert
= 2 Number aircraft required for Priority #4 alert
= 3 Pointer to first aircraft assigned to priority #2
= 4 Pointer to first aircraft assigned to priority #4
= 5 Number of aircraft readied for priority #2
= 6 Number of aircraft readied for priority #4

Squadron Equipment Equivalence Designators (INPUT-#46)

ALTAGE (NOAGE, I)

I = 1 Type designator for AGE assigned to the second squadron
= 2 Type designator for assignments to the third squadron
= 3 Type designator for assignments to the wing organization

Squadron Personnel Equivalence Designators (INPUT-#45/1)

ALTPEO (NOPEOP, I)

I = 1 Type designator for personnel assigned to the second squadron
= 2 Type designator for assignments to the third squadron
= 3 Type designator for assignments to the wing organization

Task-Assist-Qualified Personnel Types (INPUT-#45/3)

AQPEOP (NOPEOP, I)

I = 1-5 Personnel types who may assist with on-equipment tasks

Airbase Attack Data (INPUT-#40)

ATTACK (LTHATT, I)

I = 1 Attack time
= 2 Heap pointers
= 3 Heap pointers
= 4 Position of first damage data in DAMAGE array
= 5 Base

Delays due to Post-attack Disruption (GENERATED)

ATTDLY (I, MAXB)

I = 1 SHPDLY modified for attack strength
= 2 CEDELY modified for attack strength

Average Shop Performance (GENERATED)

AVGP (I, SHOP, BASE)

I = 1 Average task time
= 2 Average task capacity
= 3 Expected closure time*10/distribution

Standard Backshop Parts Repair Times (GENERATED)

AVGREP (SHOP(25), MAXT)

Ten times the average on-base repair time that would be required with unlimited resources for jobs generated by a particular aircraft type, taking into account the several probabilities affecting whether or not the job would be done on-base.

Average Inter-Base Shipment Time (GENERATED)

AVGSHP (MAXB)

Average shipment time from each operating base to all other operating bases (TTU)

Standard On-Equipment Task Times (GENERATED)

AVGTSK (SHOP(25), MAXT)

Ten times the average time that each shop would take to complete on-equipment tasks on a given aircraft type, when resources are unlimited

Munitions Buildup Wait Queue (GENERATED)

BACKLG (I, LLQ)

I = 1 Munitions type
= 2 Next task in shop (unused elements)
= 3 Resource causing delay
= 4 Time task first attempted
= 5 Base

Special Base Data (GENERATED)

BASES (I, MAXB)

| | |
|---|---|
| I = 1 | First aircraft assigned to base |
| = 2 | Last aircraft assigned to base |
| = 3 | Number of aircraft possessed |
| = 4 | Runway status (0 if open, 1 if closed) |
| = 5 | Taxiway status (runway non-access probability) |
| = 6 | Total cannibalizations |
| = 7 | Number of LRUs cross-canned |
| = 8 | Number of repairs expedited |
| = 9 | Initial number of LRUs; negative if current number less than ADAPTR percent of initial |
| = 10 | Personnel qualifications: +10 if on-base personnel are cross-trained; +1 if personnel are task-assist-qualified |
| = 11 | Number of aircraft assigned initially or by preplanned reinforcement |
| = 12 | Number of aircraft with broken or missing parts |
| = 13 | Time of last airbase attack |
| = 14 | Coded record of aircraft types assigned to base [Sum of 2**(ACTYPE-1)] |
| = 15 | Spare |
| = 16 | Parts repair organization type: 0 for data as entered; 1, when the flight line is a 66-1 organization and the data apply to a 66-5 organization (and resource equivalents must be used for repair work) |
| = 17 | Number of aircraft receiving post-flight inspection |
| = 18 | Number of aircraft that require unscheduled maintenance |
| = 19 | Ten times the average number of aircraft that may be accommodated in a shelter (default = 10) |
| = 20 | Number of shelters on base |
| = 21-29 | Number of aircraft of type (I - 20) * 8/Number of squadrons |
| = 30 | Number of COMO squadrons per aircraft type |
| = 31 | Number of serviceable parts shipped. |
| = 32 | Number of serviceable parts received from an operating base. |
| = 33 | Number of serviceable parts received from a CIRF. |
| = 34 | Number of serviceable parts received from CONUS. |
| = 35 | Number of reparable parts shipped to an operating base. |
| = 36 | Number of reparable parts shipped to a CIRF. |
| = 37 | Number of reparable parts shipped to CONUS. |
| = 38 | Number of parts condemned |
| = 39 | Current number of battle damaged aircraft |
| = 40 | Cumulative number of bent parts |
| = 41 | Base kind: 1 for MOB; 2 for COB |
| = 42 | Storage capacity for POL |
| = 43 | Number of aircraft shelters designated for "special" alert |
| = 44-45 | Spare (values read from Card Type #17) |
| = 46 | Unity if the base is used for rear-area maintenance |
| = 47 | Cumulative number of aircraft damaged in air operations |
| = 48 | Number of aircraft flown to rear for maintenance |
| = 49 | Number of aircraft transferred for filler force |
| = 50 | Spare |

Task Assistance List (GENERATED)

BORROW (NOUSER, I)

This array stores the shop numbers of shops that borrow personnel from other shops (I = 1).; or, equipment (I = 2).

Base Parts Provisioning Policy Data (INPUT-#23/70 & 23/72)

BPARTS (I, MAXT, MAXB)

I = 1 Kind of base: 1 for in-place units to have POS/BLSS
2 for deployed unit to receive a WRSK
= 2 Type of aircraft
= 3 Number of aircraft (PAA)
= 4 Peacetime sorties per day per PAA*100
= 5 Wartime sorties per day per PAA*100
= 6 Average peacetime base parts repair time (hours)
= 7 Average wartime base parts repair time (hours)
= 8 Peacetime order and ship time (days)
= 9 Wartime order and ship time (days)
= 10 One-way travel time to CIRF, when applicable (hours)
= 11 Unity when all faulty parts are to be NRTSed.
= 12 ALPHA1; safety factor for high priority LRUs
= 13 ALPHA2; safety factor for low priority LRUs
= 14 BETA1; safety factor for high priority SRUs
= 15 BETA2; safety factor for low priority SRUs

Cumulative Sorties (GENERATED)

BSOR (9)

Cumulative sorties by base

Munitions Build-up Task Heap (GENERATED)

BUILDQ (LBQ, I)

I = 1 Type of munitions
= 2 Completion time
= 3 Pointers: To heap (Unused elements)
= 4 Heap pointer
= 5 Prior task, same shop
= 6 Personnel Type*100/Number
= 7 AGE and equipment AGE1*100/AGE2
= 8 Time task first attempted
= 9 Base*100/Assembly procedure
= 10 Alternate personnel type*100/Number

Record of Pilot Shortages and Effects (GENERATED)

CANCEL (I, MAXT, MAXB)

I = 1 Cumulative number of fully ready aircraft canceled from tentative flights because of pilot shortages
= 2 Cumulative number of pilots needed to have met minimum flight size requirements
= 3-5 Spare

Sortie Generation Capabilities (GENERATED)

CANFLY (I, MAXM, MAXT, MAXB)

I = 1 Estimated daily limit without regard to available aircraft
= 2 Estimated daily limit for aircraft of the specified type that are not constrained by a "hole"
= 3 Estimated daily sortie limit for aircraft of specified type, taking into account all aircraft types on-base.

Time Required to Obtain a Part by Cannibalization (INPUT-#35)

CANNTM (NOPART)

The additional on-equipment task time required to obtain a part by cannibalization; if "-1", part may not be cannibalized. If < -1, cannibalization is permitted if more than DOCANN aircraft require this part type; the time required is the absolute value of CANNTM.

Cargo Shipped from CONUS (INPUT-#31)

CARGO (NCARGO, I)

I = 1 Base*256 + Quantity
= 2 3200*Class + Type
(64 is added to the base number for parts enroute from a CIRF to a base at zero time)

Civil Engineering Job Queue (GENERATED)

CEJOBQ (LTHCEQ, I)

I = 1 Base*256 /facility number
= 2 Personnel#1 Type*100/number
= 3 Personnel#2 Type*100/number
= 4 Equipment#1 Type*100/number
= 5 Equipment#2 Type*100/number
= 6 Task initiation time
= 7 Task completion time
= 8 Pointers: To heap (Unused elements)
= 9 Heap pointer

Civil Engineering Task Priority (INPUT-#39)

CEPRTY (NOFAC)

The facility number in the ith position is the ith on the repair priority list.

Civil Engineering Task Requirements (INPUT-#38)

CERQTS (I, NOCE)

I = 1 Time per unit task*100/time function
= 2 Personnel#1 Type*100/number
= 3 Personnel#2 Type*100/number
= 4 Equipment#1 Type*100/number
= 5 Equipment#2 Type*100/number
= 6 Material#1 Quantity*100/Type
= 7 Material#2 Quantity*100/Type
= 8 Alternate resource set

Preset and Dynamic Parameter Change Storage (INPUT-#49)

CHANGE (NOCHG, I)

I = 1 Time change is to be accomplished
= 2 Pointers: To time heap (unused elements)
= 3 Heap pointer
= 4 Type of change *100/miscellaneous data
= 5 Value after change (this number may be "packed" as required. See subroutine MODIFY).

Temporary Part Generation Status Array (GENERATED)

CHCKED (NOPART)

Set to unity when part type has been checked in subroutine IPARTS.

CIRF Parts Repair Time Modifiers (INPUT-#48)

CIRFTM (SHOP(24))

Modifies the nominal shop repair time at a CIRF by a specified percentage.

Check Filler Aircraft Assignment (GENERATED)

CKFILL (MAXT)

Automatically reset from zero to one whenever filler aircraft fall to zero; subsequently a check is made at midnight to assign any new, unassigned filler aircraft.

Aircraft Munition Configuration Data (INPUT-#14)

CONFIG (NOCONF, I)

I = 1 Task #1 - Time*100/distribution*10/personnel substitutability
= 2 TRAP Type*10/Number
= 3 Equipment EQP1*100/EQP2
= 4 Personnel Type*100/Number
= 5 Task #2 - Time*100/distribution*10/personnel substitutability
= 6 TRAP Type*10/Number
= 7 Equipment EQP3*100/EQP4
= 8 Personnel Type*100/Number

Scheduled Support Shipments from CONUS (INPUT-#31)

CONUS (NOCONS, I)

I = 1 (Day of arrival-1)*480 + Hour of arrival*20
= 2 Pointer to the position of the first item in the CARGO array

Parts Cost Data (INPUT-#23/66)

COSTS (NOPART)

The cost of an individual part (LRU or SRU) in hundreds of dollars

Temporary Parts Storage (GENERATED)

CSTOCK (NOPART, I)

I = 1 Authorized on-base stocks
= 2 Actual on-base stocks

Cross-Trained Personnel (GENERATED)

CTPEO (NOPEOP)

Entry set to unity for personnel types that are cross-trained for any activity of another specialty.

Cross-Trained Personnel Types (INPUT-#45/2)

CTPEOP (NOPEOP, I)

I = 1-5 Personnel types that may be substituted for on-equipment tasks.

Base Damage Data (INPUT-#40)

DAMAGE (NOITEM, I)

Data are packed differently for different types of resources

I = 1 Resource Class

| | | |
|---|---|---|
| Personnel | (#1) | Resource Class*3200/Type |
| AGE/Equip | (#2) | " " " " |
| Parts | (#3) | " " " " |
| Munitions | (#4) | " " " " |
| TRAP | (#5) | " " " " |
| Material | (#6) | " " " |
| POL | (#7) | " " " |
| Aircraft | (#8) | Resource Class*3200/Type*110/Percent damage to aircraft in closed regular shelters (Entry #1) or Percent damage to aircraft in closed special shelters (Entry #2) |
| Facilities | (#9) | Resource Class*3200/Facility*30/ Percent destroyed |
| Special | (#10) | Resource Class*3200 / Entry*110/ Percent regular shelters destroyed (Entry #1) or Percent damaged aircraft that are not reparable. (Entry #2) |

I = 2 Resource Class

| | | |
|---|---|---|
| #1 thru #7 | | Percent destroyed |
| #8 and #9 | | (%Peop lost)*900/(%AGE lost)*30/ (%Parts lost) |
| #10 | (Entry #1) | Percent aircraft damaged in opened regular shelters*128/Percent unsheltered aircraft damaged |
| | (Entry #2) | Percent special shelters destroyed*128/ Percent damage to aircraft in opened special shelters |

Note that there are two sets of Class #10 data, and that Class #10 data must precede Class #8 data in the input sequence. If no resource type is entered for classes 1,2,3,4,5,6, or 8 all types in class sustain same level of damage.

Deferred Task Storage Array (GENERATED)

DEFTSK (LDT, I)

I = 1 Task number
= 2 Next task, same aircraft (unused elements)
= 3 Next task, same shop
= 4 Aircraft number

Resources Available to Replace Losses (INPUT-#2x/99)

DEPOT1 (NOPEOP)
DEPOT2 (NOAGE)
DEPOT3 (NOPART)
DEPOT4 (NOMUN)
DEPOT5 (NOTRAP)
DEPOT6 (NOMATL)
DEPOT7 for POL
DEPOT8 (MAXT)

Available quantities of each type of resource that may be requisitioned to replace losses; default = 32500.

Facility Storage (GENERATED--TEMPORARY STORAGE)

FACDAM (NOFAC, I)

I = 1 Flag Set to 1 if facility damaged by attack
= 2 Percent of facility damaged
= 3 Percent personnel lost
= 4 Percent equipment lost
= 5 Percent parts lost
= 6 Spare

Base Facility Data (INPUT-#37)

FACLTY (I, NOFAC, MAXB)

I = 1 Task type for reconstruction*100
= 2 Size in units consistent with CERQTS data
= 3 Percent damage*128/Parent shop for distributed capacity
= 4 Location of repair work in CEJOBQ
= 5 Alternate shop location*128/Repair capacity
= 6 Subsequent task type*128/Task origin
= 7 Spare

Filler Aircraft (INPUT-#20/77)

FILLER (MAXT, I)

I = 1 Number of aircraft available as fillers
= 2 Time required for a filler aircraft to reach assigned base

Sortie Demand Data (INPUT-#50)

FLTRQT (MAXFLT, I)

I = 1 Launch base*128/Aircraft type*8/Mission
= 2 Priority*1000/Daily demand probability
= 3 Number aircraft required*32/Minimum number
= 4 Time flight announced before takeoff*64/Recovery base
= 5 Flight time
= 6 Pointers: Next later flight--all bases
= 7 Next earlier flight (and unused element pointer)
= 8 Next flight same mission, aircraft, base
= 9 First aircraft assigned this flight
= 10 Number aircraft assigned

Temporary Parts Allocation Array (GENERATED)

FRACBS (NOPART, MAXB)

Fraction of parts assigned to base rather than to the CIRF

Taxiway Non-access Function (INPUT-#17/1)

FTAXI (I, MAXB)

I = 1 Taxiway damage level at which the runway cannot be accessed from the shelters

= 2 Exponent controlling the variation of access probability with the level of damage

Part Replacement Time Flag (GENERATED)

GTLMT (NOPART)

Flag designating that time for associated maintenance task exceeds MNTLMT

Task Time Multipliers (INPUT-#17/2)

HURRY (MAXB, J, I)

I = 1 Nominal percentage of standard task times
= 2 Current percentage of standard task times

J = 1 Unscheduled on-equipment tasks
= 2 Preflight tasks
= 3 Off-equipment repairs
= 4 Munitions assembly jobs
= 5 Civil engineering jobs

Temporary Pipeline Parts Storage (GENERATED)

INPIPE (NOPART, MAXB, I)

I = 1 Total in pipeline consigned to base
= 2 Total in pipeline consigned off-base

Temporary Pipeline Storage (GENERATED)

IPIPE (NOPART, I)

I = 1 Total in pipeline to base
= 2 Total in pipeline for base including off-base storage

Storage Array for Interrupted Tasks (GENERATED)

INTTSK (LIQ, I)

I = 1 Task number, part or AGE repair procedure, or munition type
= 2 Basic task number (if prior is alternate)
= 3 Aircraft number or Base*64/base of origin or Base *100/assembly procedure
= 4 Pointers: Next task, same aircraft; or -LRU, when job is SRU replacement; or -SRU(+5000), for an SRU repair; or -AGE(+10000) for an equipment repair (unused elements)
= 5 Next lower priority task for shop
= 6 Next higher priority shop task
= 7 Remaining time
= 8 Time basic task initiation attempted
= 9 Time task element initiation first attempted
= 10 Root segment for elements of a task network

Numbers of Manual Entries (GENERATED)

ITEMS (MAXB)

Number of manual entries when automatic parts generation feature is used

Temporary Data Storage in Subroutine INCOMP (GENERATED)

JOBDTA (20, 2)

Cumulative Requirement for Rear Base Maintenance (GENERATED)

JOBPR (KIND, MAXT)

Cumulative probability of the tasks that must be carried out at a rear base for aircraft based at a MOB (KIND = 1)., or at a COB (KIND = 2).

Cumulative Number of Landings by Base (GENERATED)

LANDNG (MAXB)

Lateral Resupply Bases (INPUT-#23/74)

LATERL (MAXB)

Coded record of up to five bases called for lateral resupply
[ Sum of (64**N)*BASE ]

Heap for Aircraft Parts during Administrative Delays (GENERATED)

LIMBO (NLIMBO, I)

I = 1 Part number
= 2 Base*64/Original base
= 3 Time delay began
= 4 Time delay complete
= 5 To heap (Unused elements)
= 6 Heap pointer

Storage Array for Task Incompatibilities (INPUT-#19)

LISTIN (LTHLST)

This linear array is used to store task numbers, shop numbers, and blocks of task numbers that are incompatible with specific on-equipment tasks

Materials Stocks (INPUT-#26)

MATERL (NOMATL, MAXB)

Current on-base stock level for each type of material

Numbers of Part Types Required in Rear (GENERATED)

MAXOFF (KIND, MAXT)

Maximum number of part types that are required at a rear operating base for an aircraft at a MOB (KIND = 1), or at a COB (KIND = 2).

Munition Requirements per Sortie (GENERATED)

MUNRQD (NOMUN, MAXM, MAXT)

Expected requirements for munitions per sortie times 100.

Munitions Build-up Resource Requirements (INPUT-#11)

MUNRQT (I, NOBILD)

I = 1 Time*10/Distribution
= 2 Personnel: Type*100/Number
= 3 AGE: AGE1*100/AGE2
= 4 Number assembled*1000/alternate resource set*10/personnel substitutability

Ammunition Stocks (INPUT-#25)

MUNSTK (NOMUN, I, MAXB)

I = 1 Number available for loading
= 2 Number available for assembly
= 3 Total on base, except for I = 2
= 4 Temporary tally used during munitions construction

Current Number of NMC Aircraft (GENERATED)

NOR (MAXB)

Cumulative Number of NORS Hours at Each Base (GENERATED)

NORHRS (MAXB)

NOR Aircraft Storage (GENERATED)

NORQ (LNOR, I)

I = 1 Aircraft affected
= 2 Pointer to next aircraft, same item (or unused elements)
= 3 Time remaining until the ready-to-fly time at time of report

AIS Base Station Stocks

NSTAT (NOSTAT, I, MAXB)

I = 1 Total number of stations of each type on base
= 2 Number in stations in use

Parts Requirement for Rear Base Maintenance (GENERATED)

OFFBSE (KIND, 50, I, MAXT)

I = 1 Part number
= 2 Probability (*10000) that an aircraft at a MOB (KIND = 1), or a COB (KIND = 2), will require the part at a rear maintenance base.

Temporary Part Demand Storage (GENERATED)

OFFCOB (NOPART, MAXT)

Per sortie part demand probability at a COB that will be handled at a rear maintenance base

Temporary Part Demand Storage (GENERATED)

OFFMOB (NOPART, MAXT)

Per sortie part demand probability at a MOB that will be handled at a rear maintenance base

Arrays for On-Equipment Task Delay Data (GENERATED)

| | |
|---|---|
| OUTAGE (I, NOAGE, MAXB) | AGE and Equipment |
| OUTFAC (I, NOFAC, MAXB) | Facilities |
| OUTMAT (I, NOMATL, MAXB) | Building materials |
| OUTMUN (I, NOMUN, MAXB) | Munitions |
| OUTPER (I, NOPEOP, MAXB) | Personnel |
| OUTPOL (I, MAXB) | Fuel |
| OUTPRT (I, NOPART, MAXB) | Parts |
| OUTTRP (I, NOTRAP, MAXB) | TRAP |

I = 1 Incidents*1000000/sum of the delay times
= 2 Sum of the delay times squared

Sortie Production Data (GENERATED)

OUTPT1 (I, PRTY, MAXM, MAXT, MAXB)

I = 1 Cumulative sorties demanded during day
= 2 Cumulative sorties flown during day
= 3 Flight data: PRTY = 1 Demanded daily
= 2 Launched daily
= 3 Demanded overall
= 4 Launched overall
= 4 Cumulative sorties demanded during simulation
= 5 Cumulative sorties flown during simulation

Daily Shop Completion Records (GENERATED)

OUTPT2 (I, J, SHOP(25), MAXB)

I = 1 Daily number for each shop at each base
= 2 Cumulative number for each shop at each base
= 3 Spare

J = 1 On-equipment tasks
= 2 Off-equipment parts repair jobs
= 3 AGE repair jobs

Effectiveness Summaries for Sorties Flown (GENERATED)

OUTPT3 (I, MAXM, MAXT, MAXB)

I = 1 Daily total of sortie-effectiveness-proxy values
= 2 Cumulative total of these values

Overall Sortie Production Data (GENERATED)

OUTPT4 (I, J, MAXM, MAXB)

I = 1 Sorties for day J cumulated over all trials
= 2 Square of the Jth days sorties, cumulated over all trials

Shop Manhour Expenditure Records (GENERATED)

OUTPT5 (I, SHOP, MAXB)

I = 1 Cumulative manhours on on-equipment tasks by men assigned to the shop
= 2 Cumulative manhours on parts repair jobs assigned to the shop
= 3 Cumulative manhours on equipment repair jobs assigned to the shop

Shop Activity Records (GENERATED)

OUTSHP (I, SHOP, MAXB)

I = 1 Cumulative number of on-equipment tasks
= 2 Sum of total time for on-equipment tasks from the first attempt to initiate until completion
= 3 Sum of on-equipment task times squared
= 4 Cumulative number of off-equipment repair jobs
= 5 Sum of total time from first attempt to initiate repair until completion
= 6 Sum of off-equipment repair times squared
= 7 Cumulative number of AGE repair jobs
= 8 Sum of total time from first attempt to initiate repair until completion
= 9 Sum of AGE repair times squared

Parts Requirements (GENERATED)

PARTRQ (NOPART, MAXT)

Expected number of parts required per sortie ( *10000 )

Spare Parts Stocks (INPUT-#23)

PARTS (NOPART, I, MAXB)

I = 1 Number serviceables on base*100/shop number
= 2 Number reparables on base*100/total items in shop
= 3 Nominal stock level*128/percent NRTS
= 4 Pointer to NORQ array of first aircraft that requires part or -LRU
= 5 Number of aircraft requiring part*100 (or number of LRUs waiting for this SRU*100)/number serviceables enroute to an operating base (or number reparables enroute to the CIRF)

Base Personnel (INPUT-#21)

PEOPLE (NOPEOP, I, MAXB)

I = 1 Total available on base*100/'target' number
= 2 Number on "day" shift*100/'target' number
= 3 Unassigned*100/assigned off-equipment
= 4 Nominal shop*100/minimum number on shift
= 5 Number personnel enroute to base
= 6 Shift change status: =1 when checked; =2 all released
= 7 Number remaining to be released after shift change

Resource Report on Personnel (GENERATED)

PEORPT (I, NOPEOP, MAXB)

I = 1 Data in transit for total personnel on base
= 2 Data received for total personnel on base

Personnel Requirements (GENERATED)

PEORQT (NOPEOP, MAXM, MAXT)

Likelihood needed*(10**7)/Expected requirements for personnel per sortie (10000 * men * TTU).

Periodic/Scheduled Task Time Heap (GENERATED)

PERIOD (I, J)

I = 1 Planning and shift changes
= 2 Next flight schedule input time
= 3 Next time for scheduling flights, if none input
= 4 Daily summary of results
= 5 Next time for periodic resource management
= 6 Schedule intra-theater shipments
= 7 Receive shipments from CONUS
= 8 Next shipment departure
= 9 Next shipment arrival
= 10 Transmit and receive reports
= 11 Periodic "hole" summary
= 12 Conclude administrative parts delays
= 13 Periodic computation of base capabilities
= 14 Next parameter change
= 15 Next airbase attack
= 16 Next special report of deferred tasks
= 17-20 No spare

J = 1 Time of earliest event
= 2 Pointers: To time heap
= 3 Heap pointer

Air-Crew Status Data (GENERATED)

PILOT (I, NOCREW)

I = 1 Pointer: Next crew at rest, same aircraft type
= 2 Next crew on-duty, same aircraft type
= 3 Earliest time off-duty period complete or Time on-duty
= 4 Landing time most recent flight
= 5 Tentative assignment flag

Air-Crew Locator Data (GENERATED)

PILOTS (I, MAXT, MAXB)

I = 1 Number of aircrews on-base
= 2 Pointer: First air-crew assigned to rest
= 3 Last air-crew assigned to rest
= 4 First air-crew assigned to a flight
= 5 Last air-crew assigned to a flight

On-Base Parts Repair Policy Data (INPUT-#23/2xx & #23/3xx)

POLICY (NOPART, MAXB, I)

I = 1 The NRTS rate for each part at each base when there is no CIRF*100
= 2 The NRTS rate for each part at each base when there is a CIRF*100

Base Fuel Stocks (INPUT-#27)

POLSTK (MAXB)

On-base fuel stocks.

Periodic Flight Data Storage (INPUT-50)

PRDFLT (MAXFLT, I)

I = 1 Launch base*128/aircraft type*8/mission
= 2 Priority*1000/Daily demand probability
= 3 Number flights required*1024/Number aircraft required*32/minimum number of aircraft that are acceptable
= 4 Time flight announced before takeoff(hr)*16/Recovery base
= 5 Launch time uncertainty (min/10)*512/Daily launch time

Part Criticality Data (GENERATED)

PRTCRT (NOPRT, 2)

Provides a record of the criticality of each part for up to 9 types of aircraft for which it may be used and for each mission that that aircraft type may fly

Aircraft Parts Lists (INPUT-#28)

PRTLST (NOPRT)

Entries are part number*10/number installed on each aircraft

Resource Reports on Parts (GENERATED)

PRTRPT (I, NOPART, MAXB)

I = 1 Data in transit regarding number of usable parts
= 2 Data received regarding number of usable parts
= 3 Reparables on base--Data received*128/Data in transit
= 4 Number aircraft NORS--Data received*128/Data in transit

Temporary Parts Demand Data (GENERATED)

PRTRQ (NOPART, I, MAXT)

Temporary storage array for accumulating demand for a part needed in a task network after parallel paths have split and rejoined (see subroutine CKSPLT).

I = 1 Cumulative probability part is required on mutually exclusive paths
= 2 Cumulative probability part is required on non-mutually exclusive paths

Flight Requirements Pointers (GENERATED)

PTZ (MAXM, MAXT, MAXB)

Pointer to location of first sortie demand of a specific type

Daily Aircraft Activity Storage Array (GENERATED)

RECORD (24, I, MAXREC)

I = 1 Time of day for completion of task
= 2 Time of day task was initiated
= 3 Task number; zero designates a sortie; -1, a lost aircraft

Task Time Reduction Factors (INPUT-#17/2)

REDUCE (MAXB, J, I)

I = 1 Nominal reduction in TTU in standard task times
= 2 Current reduction in TTU in standard task times

J = 1 Unscheduled on-equipment tasks
= 2 Preflight tasks
= 3 Off-equipment repairs
= 4 Munitions assembly jobs
= 5 Civil engineering jobs

Theater Resource Requisition Control Data (INPUT-#43)

REFILL (I, J)

I = 1 Switch*100/Time distribution
= 2 Mean resupply time

J = Resource Class

Temporary Split Rejoin Records (GENERATED)

REJOIN (NJOINT, I)

Maintains record of parallel paths that have not yet rejoined

I = 1 Task element where paths rejoin
= 2 Pointer to next path location that rejoins (unused elements)

Relative Importance of On-Equipment Tasks (ENCODED)

RELIMP (33, MAXM)

Stores, for each task criticality index, the number of missions for which task is critical.

Alternative Parts Repair Procedures (INPUT-#9)

REPALT (NOREPA, I)

I = 1 Required time*10/distribution
= 2 Personnel Type*100/number
= 3 AGE AGE1*100/AGE2
= 4 Alternate resource set

Daily Base Resource Reporting Schedules (INPUT-#36)

REPORT (NOREPT, I)

I = 1 Transmittal or receipt time (20*HR+MIN/3)
= 2 Heap pointers
= 3 Heap pointers
= 4 Base

Storage Queue for In-process Parts Repair (GENERATED)

REPQ (LRQ, I)

I = 1 Part or equipment repair resource set
= 2 "Basic" resource set (if prior an alternative)
= 3 Base*64/Base of origin
= 4 Completion time
= 5 Pointers: To time heap (Unused elements)
= 6 Heap pointer
= 7 Prior repairs, same shop
= 8 Resources: Personnel Type*100/Number
= 9 AGE AGE1*100/AGE2
= 10 Parent LRU, for an SRU replacement job; SRU(+5000), for an SRU repair; or, AGE(+10000), for an equipment repair
= 11 Time job initiation first attempted

Basic Parts Repair Procedures (INPUT-#8)

REPRQT (NOREP, I)

(Data content varies for parts with one or more types of repair, for an LRU and for SRUs)

| | Simple Part with single repair procedure | SRU or other repair procedure | LRU or part with multiple repair procedures |
|---|---|---|---|
| I = 1 | Shop | Next procedure or SRU | Shop |
| = 2 | Mean repair time*10/distribution | | Expected time*10 |
| = 3 | Personnel: | Type*100/Number | -1 for LRU<br>-2 for multi-task |
| = 4 | AGE: | AGE1*100/AGE2 | First procedure or SRU |
| = 5 | --- Alternative resource set --- | | --- |
| = 6 | MTBF | Probability*100 | MTBF |
| | (where MTBF is expressed as sorties per failure and is generated in subroutine AVGTSK) | | |
| = 7 | | First SRU repair procedure; or, -1 for a procedure rather than an SRU | |
| = 8 | Spare | | |

Requisitioned Resource Storage Heap (GENERATED)

RESUPP (LGQ, I)

| | | |
|---|---|---|
| I = 1 | Base*256 + Number | |
| = 2 | 3200*Class + Type | |
| = 3 | Arrival time | |
| = 4 | Pointers: To time heap | (Unused elements) |
| = 5 | Heap pointer | |

Storage of Readiness Data (GENERATED)

RINDEX (I, MAXB)

| | |
|---|---|
| I = 1 | Cumulative number of aircraft readied for flight in two hours |
| = 2 | Cumulative number of aircraft readied for flight in four hours |
| = 3 | Cumulative number of aircraft readied for flight in six hours |
| = 4 | Cumulative number of aircraft readied for flight in eight hours |

Part Location in Task Network Structure (GENERATED)

ROOTS (NOPART, MAXT)

Entry is task network root element for network within which part is located.

Temporary Storage of Mandatory Aircraft Tasks (GENERATED)

RQDTSK (LRT, I)

I = 1 Task number
= 2 Pointer to next task, same ac (or unused elements)

Task Time Save Factors (INPUT-#17/2)

SAVE (MAXB, J, I)

I = 1 Nominal reduction in overall task times in TTU
= 2 Current reduction in overall task times in TTU

J = 1 Unscheduled on-equipment tasks
= 2 Preflight tasks
= 3 Off-equipment repairs
= 4 Munitions assembly jobs
= 5 Civil engineering jobs

SCL Preference Listing (INPUT-#12)

SCLP (Priority(5), MAXM, MAXT, I)

I = 1 Aircraft combat loading in order of preference for each aircraft and mission
= 2 Mission-SCL sortie effectiveness proxy

Resource Requirements for Loading SCLs (INPUT-#13)

SCLRQT (NOSCL, I)

I = 1 Configuration*10/Flag; shop required if unity
= 2 Required time for first munitions*100/distribution*10/personnel substitutability
= 3 #1 Type ammunition Type*100/Number
= 4 #1 Loading equipment EQP1*100/EQP2
= 5 #1 Personnel required Type*100/Number
= 6 Required time for second munitions*100/distribution*10/personnel substitutability
= 7 #2 Type ammunition Type*100/Number
= 8 #2 Load equipment EQP3*100/EQP4
= 9 #2 Personnel required Type*100/Number

Random Number Stream Control (INPUT-#2/2)

SEEDED (10)

Only when reset by entries on the #2/2 Card are the controlled random number streams disengaged.

Seeds for the Controlled Random Number Generators (ENCODED)

SEEDS (10)

Ten seeds are stored for random number streams that may be repeated from trial to trial (see SEEDED).

Actual Intra-Theater Shipping Schedules (GENERATED)

SHIP (NOSHIP, I)

I = 1 Shipment number (i.e., position in SHIPSC array)
= 2 Departure time
= 3 Arrival time
= 4 Pointers: Next departure, same base
= 5 Next departure, all bases
= 6 Next arrival, all bases
= 7 SHIPQ location of first resource in shipment

Intra-Theater Shipment Storage (GENERATED)

SHIPQ (NOPKG, I)

Unit quantities of the various resources must be defined such that the "quantity shipped" is never as large as 100.

I = 1 Base of origin *256 + Quantity
= 2 3200*Resource Class + Type of resource
= 3 Pointer to next item, same origin, same destination, and same shipment (unused elements)

Nominal Shipping Schedules (INPUT-#32)

SHIPSC (NOSHP, I)

I = 1 Originating base*64/Destination
= 2 Last day scheduled*100/Departure frequency (days)
= 3 Nominal departure hour

Nominal Transportation Time Delays (INPUT-#33)

SHIPTM (Origin, Destination, I)

I = 1 Takeoff delay*16/time distribution
= 2 Enroute time*16/time distribution
= 3 Probability of arrival*100 (is set negative when no shipment schedule is active for origin and destination)

Parts Shipping Instructions (INPUT-#34)

SHIPTO (MAXB, J, 2)

This array stores up to J (20) pairs of data for each base; the first number of each pair is a part number, and the second is a base number. If the first two pairs (i.e., for J = 1 and 2) were 10:2 and 30:5, that would signify that parts 1 through 10 should be sent to Base #2 for repair, and that parts 11 through 30 should go to Base #5.

Temporary Shop Equipment Storage Array (GENERATED)

SHOPAG (NOAGE)

Used to store damage data during air base attack computations.

Temporary Shop Personnel Storage Array (GENERATED)

SHOPEO (NOPEOP)

Used to store damage data during air base attack computations.

Shop Facility Requirements for On-Equipment Tasks (GENERATED)

SHOPRQ (SHOP, MAXM, MAXT)

Average probability that shop facility is required per sortie.

Shop Activity Status Array (GENERATED)

SHOPS (I, SHOP, MAXB)

I = 1 Number of on-equipment tasks in process
= 2 Number of parts repair jobs in process (or minus percent damage)
= 3 Pointers: First interrupted task
= 4 Last interrupted task
= 5 Number of interrupted tasks
= 6 First waiting task
= 7 Last waiting task
= 8 Number of tasks waiting
= 9 First task in TASKQ
= 10 Last task in TASKQ
= 11 First interrupted repair
= 12 Last interrupted repair
= 13 Number of interrupted repairs
= 14 First waiting repair
= 15 Last waiting repair
= 16 Number of repairs waiting
= 17 First repair in REPQ
= 18 Last repair in REPQ
= 19 Hour (even-numbered) that day shift begins
= 20 Pointer to first shop that borrows personnel*128/Percent of tasks for which the aircraft is partially exposed while in a shelter
= 21 First deferred task
= 22 Last deferred task
= 23 Pointer to first shop that borrows AGE
= 24 Current job capacity at distributed shop locations (default=10000)
= 25-26 Spares

Part Shortage Percentage (GENERATED)

SHORT (NOPART)

Temporary storage array in IPARTS.

Shop Sequence Control Data Array (INPUT-#29)

SHPORD (50, MAXT, MAXB)

A zero separates simultaneous sets of tasks and shops; two zeros end the sequence.

Shipment Pointers (GENERATED)

SHPT (Origin, Destination, I)

I = 1 Location of the first shipment in the SHIP array
= 2 Location of the last shipment in the SHIP array
= 3 Pointer to the position in the SHIPQ array of first item without scheduled transport

Shop Task Probability Storage (INPUT-#30)

SHPTSK (I, NOTASK, SHOP(25), MAXT)

I = 1 Cumulative task probability as input for planning
= 2 Task number
= 3 Cumulative task probability as used for simulation (see UNCER)

Sortie Generation Capabilities (GENERATED)

SORCAP (MAXT, MAXB)

Rough estimate of the daily number of sorties that can be flown.

Sortie Priority and Deficiency Data (GENERATED)

SORDEF (16, I, MAXM, MAXT, MAXB)

Data for sixteen time-blocks from the last even-numbered hour.

I = 1 Highest priority with a projected deficiency*1000/Remaining demand for sorties
= 2 Projected deficiency at highest deficient priority
= 3 Projected supply at the highest deficient priority

Hourly Record of Daily Sorties (GENERATED)

SORTHR (24, MAXB)

Total sorties launched each hour without abort during the current day.

Aircraft Spares (GENERATED)

SPARE (I, MAXM, MAXT, MAXB)

I = 1 Pointer to first spare aircraft
= 2 Number of spare aircraft

Temporary Personnel Storage Array (GENERATED)

STAFF (NOPEOP, I)

Stores preattack personnel levels in subroutines BOMB and REORGN.

I = 1 Total on-base
= 2 Number unassigned on-duty personnel

Time Intervals for Debug Data (INPUT-#2)

START (6)
STOP (6)

The beginning and end of six time intervals during which the debug output is to be printed.

Temporary Data Storage for Composite Flights (GENERATED)
SVEFLT (I, 5)
I = 1 Total assigned to the composite flight
= 2 Aircraft assigned to the component flight
= 3 Component flight number
= 4 Mission
= 5 Aircraft type
= 6 Base
= 7-11 Misc. factors

Aircraft In-process Tasks Storage Array (GENERATED)

TASKQ (LTQ, I)

I = 1 Task number
= 2 "Basic" task number (when prior is an alternate)
= 3 Aircraft number
= 4 Completion time
= 5 Pointers: To time heap (unused elements)
= 6 Heap pointer
= 7 Next task, same aircraft
= 8 Prior task, same shop
= 9 Resources: Personnel - Team1 - Type*100/Number
= 10 AGE AGE1*100/AGE2
= 11 Time basic task initiation attempted
= 12 Time task element initiation first attempted
= 13 Root segment for elements of a task network
= 14 Additional personnel on Team1 - Type*100/Number
(negative for a load crew)
= 15 Personnel Team2 - Type*100/Number
= 16 Additional personnel on Team2 - Type*100/Number

Nominal Reconfiguration Times (GENERATED)

TCONF (MAXM1, MAXM2, MAXT)

MAXM1 Next mission
MAXM2 Prior mission

Nominal time to reconfigure an aircraft from the preferred configuration for one mission to that for another

Temporary Data Storage for Flight Aircraft (GENERATED)

TEMPF (50, I)

I = 1 Assigned aircraft
= 2 Previously assigned aircraft
= 3 Assignment
= 4 Crew number

Planning Time-Horizon Data (INPUT-#3/2)

THDATA (J, I)

I = 1 Horizon data (I=2,3) applies when time of day is greater than THDATA (J-1,1) and no more than THDATA (J,1)
= 2 Time horizon (TTU)
= 3 Length of the 16 time blocks within the time horizon (TTU)

J = 1-4 Provides for four different time horizons for planning

Temporary CIRF Pipeline Parts Storage (GENERATED)

TOCIRF (NOPART, 1)

I = 1 Total in CONUS-CIRF pipeline stock (for CIRF and bases)
= 2 Portion of CONUS-CIRF pipeline to be retained at CIRF

Temporary Parts Storage (GENERATED)

TOTALS (NOPART, MAXB, I)

I = 1 Authorized numbers of parts
= 2 Actual on-base numbers of parts
= 3 Actual numbers of parts allocated to a CIRF

Temporary Parts Storage Array (INPUT-#23)

TPART (EXTPRT, I, ,MAXB)

Used with the automatic parts initialization feature to temporarily store addition.al stock information.

I = 1 Number serviceables on base *100
= 2 Number reparables on base *100/total items in shop
= 3 Nominal stock level*128/percent NRTS
= 4 Part number

TRAP Stock Data (INPUT-#24)

TRAP (NOTRAP, MAXB)

Current on-base stock level for each type of TRAP.

Requirements for Expendable TRAP

TRAPRQ (I, 3, T)

I = 1 TRAP type (only three types per aircraft type)
= 2 Expected number of expendable TRAP required per sortie

Parts Repair Procedure (INPUT-#23/78)

TRAY (NOPART)

AIS tray number used to repair part.

AIS Tray Characteristics (INPUT-#22/77)

TRAYS (NOTRAY)

Probability that a particular tray is affected by the non-availability of an AIS component *10000.

AIS Tray Status Data (GENERATED)

TRAYST (NOTRAY, I, MAXB)

I = 1 Unity if tray at station #1 is out of service
= 2 Pointer to next affected tray

Sortie Demand Summary (GENERATED)

TRYFLY (6, MAXT, MAXB)

Daily tally of the sorties demanded during each of six 5-hour periods starting at 2000.

Alternative Aircraft Task Procedures (INPUT-#7)

TSKALT (NOTSKA, I)

I = 1 Required time*10/distribution
= 2 Personnel required - Team1 - Type*100/Number
= 3 AGE required AGE*100/AGE2
= 4 Alternative resource/Shop required if >0
= 5 Personnel required - Team2 - Type*100/Number

On-Equipment Task Criticality (ENCODED)

TSKCRT (Task Criticality Index(99), 5)

For each value of the task criticality index, stores a coded number that denotes whether the task is essential for each of the five different mission types. A zero denotes that the task is not essential, a one denotes that it is. TSKCRT is initialized in BLOCK DATA.

Total On-Equipment Task-Incidence Probability (INPUT-#44)

TSKPR (SHOP(25), MAXT, I)
I = 1 The cumulative, per sortie probability that an aircraft of a specified type will generate a problem that will (eventually) require shop attention; value used for planning
= 2 Percentage that modifies the breakrates for each task in a given shop for a specified aircraft type*128/percent reduction in breakrate per sortie/day/PAA achieved above unity when VBREAK = 1
= 3 As for I = 1, except value is that used for simulation

Basic Aircraft Task Procedures (INPUT-#5)

TSKRQT (NOTSK, I)

I = 1 Nominal shop*10/Coded entry designating repair location and shop requirement (see Vol. II, Fig. 4)
= 2 Part number for root segment; with task following root segment, task will be skipped when -1 is entered
= 3 Time required*10/distribution
= 4 Personnel required - Team1 - Type*100/Number
= 5 AGE required AGE1*100/AGE2
= 6 Alternative resource set
= 7 Parallel task
= 8 Subsequent task
= 9 Probability (in tenths of percent) task is required*10/Flag where Flag is defined as:
1,3,5 or 7 if cross-trained personnel may be used;
2,3,6 or 7 if task-assist-qualified personnel may be used;
4-7 if the task is unscheduled maintenance
= 10 Expected total time for network
= 11 Pointer to first incompatible task
= 12 Probability part is required*128/Probability part is condemned
= 13 Task criticality*64/Task stress
= 14 Personnel required - Team2 - Type*100/Number
= 15 Split-rejoin flag

Storage Array for Waiting Tasks (GENERATED)

WAITSK (LWQ, I)

1 = 1 Task number, or part, SRU, or equipment repair procedure
= 2 Aircraft number or (Base*64/base of origin)
= 3 Number of part required, if any; or preflight status flag *10/ personnel substitutability
= 4 AGE for on-equipment tasks; LRU, when SRU replacement job is waiting; SRU(+5000), when SRU repair waiting; or AGE(+10000), when selected AGE procedure waiting
= 5 Personnel for on-equipment task; or, for repairs, SFLAG (=1 when required SRU has been set aside)
= 6 Pointers: Next task, same aircraft (unused elements)
= 7 Next lower priority task in shop
= 8 Next higher priority shop task
= 9 Estimate of time remaining before aircraft ready to fly
= 10 Resource causing wait; Class*3200/Type
= 11 Time basic task initiation attempted, or repair began administrative delay
= 12 Time task element initiation first attempted, or repair initiation was first attempted
= 13 Root segment for elements of a task network

Theater Weather (INPUT-#30)

WXDATA (DAY, GROUP(2), MAXB)

The five-digit number stored in each element is packed. Each of the two groups of numbers apply to a subset of the aircraft types. The left-hand digit of the first and second groups denote the flying conditions for aircraft types #1 and #6, respectively; subsequent digits refer to the other aircraft types in numerical order. A zero denotes that the conditions are flyable, a 1 that they are not. DAY may not exceed WXDAYS.

Resource Report Transmittal Data (INPUT-#36)

XMIT (I, MAXB)

I = 1 Transmittal time [30*(20*HR+MIN/3)]/Distribution
= 2 Loss rate of individual data*100
= 3 Loss rate for entire report*100
= 4 Base communications status--Link closed if unity

Multi-trial Base Statistics (GENERATED)

XSTAT (10, 30, MAXB)

Miscellaneous output statistics by day and base.

Multi-trial Theater Statistics (GENERATED)

XXSTAT (10, 30)

Miscellaneous overall output statistics

Zero-time Parts Activity List (GENERATED)

ZPRTRQ (NOPART)

Used to store a specially constructed parts list required in initializing the zero-time shop activities.

Initialization of Maintenance Activity at Zero Time (INPUT-#42)

ZTASKS (I, MAXT, MAXB)

I = 1-3 Percent aircraft with ongoing tasks at time zero*100/Number of tasks (a 3-part distribution)
= 4 Number of parts in administrative delay at time zero
= 5 Number of parts repairs at time zero

# APPENDIX C

## TSAR SUBROUTINES AND PRIMARY FUNCTION

The complete FORTRAN source code for the TSAR airbase simulation is organized into ten functionally related groups of subroutines that have normally been filed in ten sections as Tab A through Tab J. The general contents of each tab are indicated below, and the names and basic functions of each of the subroutines in each group are listed on the following pages in the order in which they are filed. Definitions of the primary variables and data storage arrays will be found in Appendixes A and B. All subroutine statements, entry statements, and call statements are listed, along with their argument lists, in the order that they appear in Appendix E.

| | |
|---|---|
| TAB A | Simulation Management |
| TAB B | Input |
| TAB C | Initialization and Output |
| TAB D | Sortie Demand and Aircrew Management |
| TAB E | Aircraft Maintenance |
| TAB F | Aircraft Preflight Maintenance |
| TAB G | Parts Repair |
| TAB H | Airbase Attack and Recovery |
| TAB I | Communications |
| TAB J | Support Services |

Subroutine Organization and Primary Function

TAB A Simulation Management

| | |
|---|---|
| MAIN | Executive |
| TRIALS | Manage Trials |
| MANAGE | Manage Simulation |
| ADAPT | Manage Adaptive Behavior |
| CONTRL | Manage Theater Resources |
| FERRY | Recover, Transfer, and Divert Aircraft |
| GOHOME | Manage Aircraft at Emergency Recovery Base |

TAB B Input

| | |
|---|---|
| INIT | Manage Block Storage |
| INPUT | Enter Airbase Resource Data |
| INPUTA | Aid INPUT to Read and Store Card Types #6-39 |
| INPUTB | Aid INPUT to Read and Store Card Types #41-49 |
| INPUTC | Read Attack and Damage Data from Card or Disk |
| WRAPUP | Control Initialization Data Manipulation |
| ICHECK | Check and Record Shops that Borrow Personnel/AGE |
| HELPCK | Assist ICHECK |
| INLIST | List Specified Data Arrays |
| HEADER | List Summary of Simulation Basic Conditions |
| INITIZ | Initialize Heaps, Queues, and Aircraft |
| ZSHOPS | Initialize On- and Off-equipment Activity |
| TESTER | Edit Card Input Data |
| MODIFY | Manage Time Dependent Parameter Changes |
| CKNET | Check Task Network Segments |
| CKRQT | Determine Parts Requirements for CKNET |
| CKSPLT | Assist CKNET with Split and Rejoin Networks |

TAB C Initialization and Output

| | |
|---|---|
| COMPRT | Control Spare Parts Initialization |
| IPARTS | Initialize Base Parts Stock |
| IPART2 | Assist IPARTS to Initialize Parts Pipeline |
| CKNRTS | Compute Effective NRTS Rates |
| RREQTS | Compute Average Resource Demand Data |
| REQTS1 | Assist RREQTS |
| OUTPUT | List Daily, Trial, and Overall Results |
| ASSETS | List Current Stock Levels |
| ASSET2 | List Current Stocks Levels by Type |
| TIMES | Collect Task Time Data |
| DELAYS | Prepare and List Task Times and Delays |
| PSHORT | Estimate Parts Shortages |
| JOBLST | Format and Print Aircraft Time Histories |
| DEFERS | Compute and Print Deferred Task Summaries |

TAB D — Sortie Demand and Aircrew Management

| | |
|---|---|
| READFT | Enter Sortie Demand Data |
| FRAG | Select Base For Sortie Demand |
| PLAN | Project Sortie Supply |
| PLAN1 | Project Sortie Demand and Deficiencies |
| BASCAP | Estimate Base Capabilities to Generate Sorties |
| REASSG | Revise Assigned Mission |
| FLYERS | Manage Aircrews |
| DISABL | Eliminate Lost Air Crews |
| FLIGHT | Assemble Ready Aircraft and Crews |
| LAUNCH | Launch Flights |
| SORT | Order Launch Schedules |

TAB E — Aircraft Maintenance

| | |
|---|---|
| PSTFLT | Assess Maintenance Requirements, Designate Tentative Mission Assignments |
| RUNAC | Manage Aircraft Maintenance |
| INITSK | Check Resource Availability to Initiate Tasks |
| DOTASK | Enter Tasks into In-process Heap |
| ENDTSK | Conclude On-equipment Tasks, Release Resources |
| CHKWX | Check Weather for Deferred Maintenance |
| INIDEF | Manage Deferred Aircraft Maintenance |
| CANNIB | Select Donor Aircraft for Parts Cannibalization |
| INCOMP | Check for Task Incompatibilities |
| CKCRIT | Assist PSTFLT in Assessing Ready-to-fly Time |
| CKROOT | Prevent Multiple Processing of Chained Jobs |
| SCHJOB | Organize Tasks for Aircraft Ferried to Rear |
| SPLIT | Manage Network Paths that Split and Rejoin |
| GETPEO | Locate Personnel for On-Equipment Tasks |

TAB F — Aircraft Preflight Maintenance

| | |
|---|---|
| PREFLT | Manage Preflight Maintenance |
| ASSIGN | Finalize Aircraft Mission Assignment |
| RECNFG | Check and Perform Needed Reconfiguration |
| UPLOAD | Load Munitions |
| REFUEL | Refuel Aircraft |
| DOWPRE | Check and Initiate Waiting Preflight Tasks |
| MUNEED | Establish Munitions Requirements |
| CKBILD | Define Munitions Assembly Requirements |
| DOBILD | Initiate and Complete Munitions Assembly |
| CKPEOP | Check for Personnel Substitutions |
| CKAGE | Check AGE Requirements |
| ADDAGE | Reorganize Equipment for a COMO Organization |
| CKALRT | Manage Resources Required for an Alert Aircraft |
| RELALT | Release Alert Aircraft Resources |

TAB G — Parts Repair

| | |
|---|---|
| ADMIN | Receive Faulty Parts and Manage Administrative Delay Heap |
| RUNSHP | Manage Disposition of Repaired Parts |

| | | |
|---|---|---|
| | INIREP | Check Resource Availability to Initiate Repairs |
| | DOREP | Enter Repairs into In-process Heap - REPQ |
| | ENDREP | Conclude Repairs, Release Resources |
| | SALVAG | Disassemble LRUs to Provide SRUs for Repair |
| | CKCIRF | Reassign CIRF Resources |
| | REPRTY | Establish CIRF Repair Priorities |
| | CKAIS | Manage AIS Activity |
| | NRTSIT | Select Location to Receive Reparables |
| TAB H | | Airbase Attack and Recovery |
| | BOMB | Inflict Specified Damage |
| | REORGN | Reorganize Base Operations |
| | REORG2 | Complete Base Reorganization |
| | PICK | Locate Activity in Distributed Shop |
| | REBILD | Manage Post-attack Reconstruction |
| | INICON | Assign Resources and Initiate Reconstruction |
| | BSEREP | Conclude Base Repairs, Release Resources |
| | ENDAC | Eliminate Records for Aircraft Killed On Base |
| | KILLAC | Eliminate Aircraft |
| | FTIME | Compute Reconstruction Time |
| | SHCIRF | Ship Faulty Parts to CIRF When Shop Damaged |
| TAB I | | Communications Systems |
| | SCSHIP | Schedule Intra-theater Shipments |
| | SHPRES | Prepare Resources for Shipment |
| | ORDER | Order Replacement Resources from CONUS |
| | DOSHIP | Manage Departures and Arrivals |
| | STATUS | Transmit and Receive Resource Status Reports |
| TAB J | | Support Services |
| | SHIFT | Manage Shift Changeover |
| | REDPEO | Reduce Staff Level and Reorganize Shifts |
| | CHECK | Check Requirements for Released Resources |
| | STRTSK | Store Required and Deferred Tasks |
| | REPNOR | Enter and Remove Aircraft "Hole" Reports |
| | AVGTME | Estimate Unconstrained Shop Performance |
| | INTRUP | Manage Time-ordered Interrupted Queues |
| | WAIT | Manage Time-ordered Wait Queues |
| | HEAP | Manage Data Heaps |
| | ACWAIT | Insert On-Equipment TAsks into WAITSK Array |
| | RESET | Reset Event Times for Extended Simulations |
| | BLOCK DATA | Store Task Criticality Definition Data |
| | TTIME | Select True Time from Distributions |
| | SHPRQT | Select Unscheduled Maintenance Tasks |
| | BREAK | Compute Variable Breakrate Factors |
| | LOSSES | Sample Binomial Loss Distribution |
| | RANDG | Generate "Controlled" Random Numbers |
| | ACCRIT | Compute Aircraft Criticality Periodically |

Minor Functions
LIST1, LIST2, LIST3, LIST4, LIST5
THF, TH, TOD, DATE, DAY, HRMIN, SHOPST

HELPER Assistant for Debugging

ENTRYP Assembly Language Routine to Define Origin in Core During Execution

Auxiliary Programs

SIZE.TSAR.STORAGE Compute Required Storage Space for TSAR's Labeled Common Statements

## APPENDIX D

## LOAD MODULE MAP

This appendix reprints portions of the load module map generated during the link-edit step of a compile and link-edit task. The name of each subroutine and the name of entry points in each subroutine are listed. In addition the size of the storage area required for each subroutine (expressed in terms of the number of 8-bit bytes required) and the location of the subroutine in the overlay structure are given. The sizes listed are consistent with the illustrative values used in Appendix H.

| | LENGTH | ENTRY POINTS | | | |
|---|---|---|---|---|---|
| **Subroutines in the Root Segment** | | | | | |
| MAIN | 220 | | | | |
| TRIALS | 6D2 | | | | |
| MODIFY | C3A | NEWVAL | ZVALUE | | |
| CKNET | 1354 | CKDAM | | | |
| HEAP | C10 | INHEAP | OUTHEP | EXHEAP | MODHEP |
| SHPRQT | 6C4 | SHPRQ | | | |
| TTIME | 626 | | | | |
| RANDG | 3A4 | IRANDG | CRANDG | | |
| SHOPST | 13E | | | | |
| DATE | 150 | | | | |
| DAY | DC | | | | |
| DATIME | 2F7 | | | | |
| TOD | E6 | | | | |
| HRMIN | 10A | | | | |
| THF | 15A | | | | |
| TU | 15A | | | | |
| **Common Statements in the Root Segment** | | | | | |
| BASIC1 | 2044 | | | | |
| BASIC2 | 37F8 | | | | |
| BASIC3 | 3AF0C | | | | |
| BASIC4 | 550 | | | | |
| STOCKS | 1E996 | | | | |
| JOBS | 16F6A | | | | |
| LOAD | 2EE0 | | | | |

| | |
|---|---|
| THEATR | 82C8 |
| BOMBSE | 6ADC |
| REQTS | 5894 |
| INFO | 11B1A |
| OUT | 1D5C0 |
| CPARTS | 7FDE |
| AISCOM | 1A80 |
| TESTS | 1C |
| ZZNORS | C8 |
| TIMHOR | 18 |
| RECNF | C |
| LDAMMO | 16 |
| TEMP1 | 12 |
| TEMP2 | 2 |
| TEMP3 | 12 |
| TEMP4 | 24 |
| TEMP5 | BB8 |

ENTRY POINTS

Subroutines in the Input and Initialization Segment

| | | | | | |
|---|---|---|---|---|---|
| INIT | 1056 | STORE | RECALL | DOSAVE | RECOVR |
| INPUT | 1FF0 | | | | |
| INPUTA | 405C | | | | |
| INPUTB | FA2 | | | | |
| INPUTC | DA2 | INPUTD | INPUTE | | |
| TESTER | 2588 | | | | |
| WRAPUP | 2768 | | | | |
| ICHECK | 22A8 | | | | |
| HELPCK | 832 | IHELPC | | | |
| INLIST | 1BE6 | | | | |
| HEADER | 1706 | | | | |
| INITIZ | ABE | | | | |
| CKSPLT | 8B6 | TEMPRQ | ZSPLIT | | |
| COMPRT | 105C | | | | |
| IPARTS | 4BDA | | | | |
| IPART2 | 329A | | | | |
| CKNRTS | 75C | | | | |
| PSHORT | 21E | | | | |
| RREQTS | D7A | | | | |
| REQTS1 | 114A | REQTS2 | REQTS3 | | |
| AVGTME | 17E2 | | | | |
| ENTRYP | A6 | | | | |
| CKRQT | B2A | CKRQMT | CKRQT2 | | |
| LIST1 | 4B2 | LIST1E | | | |
| LIST2 | 20C | | | | |
| LIST3 | 22C | | | | |
| LIST4 | 4AA | LIST4E | | | |
| LIST5 | 20C | | | | |

Common Statements in the Initialization Segment

| | |
|---|---|
| PURGE1 | 9948 |
| PURGE2 | 6D60 |
| PURGE3 | 29850 |
| LOCAL1 | 4F4 |

ENTRY POINTS

Subroutines in the Simulation Segment

| | | | | | |
|---|---|---|---|---|---|
| MANAGE | 2BAE | MANAG | | | |
| CONTRL | 3296 | SEND | ORDERP | GETPRT | FINDPT |
| ADAPT | 50C | | | | |
| FERRY | 2EB8 | LAND | NEWAC | | |
| GOHOME | 45A | | | | |
| | | | | | |
| ZSHOPS | 157E | | | | |
| ZNOR | 6E2 | | | | |
| | | | | | |
| OUTPUT | 4714 | | | | |
| ASSETS | A60 | | | | |
| ASSET2 | 76E | | | | |
| TIMES | 6B6 | DOTIME | WAITS | | |
| DELAYS | 3D66 | DELAY1 | DELAY2 | | |
| JOBLST | 61A | | | | |
| READFT | 1EDC | DAYONE | SCHFLT | SORTIE | |
| FRAG | DCA | | | | |
| PLAN | 1AC6 | | | | |
| PLAN1 | D14 | | | | |
| BASCAP | 3BAA | | | | |
| REASSG | 1EB0 | | | | |
| FLYERS | 12E6 | GETPLT | SAVPLT | FLYAC | LANDAC |
| | | RELIEF | | | |
| DISABL | 72A | | | | |
| FLIGHT | 2F5A | | | | |
| LAUNCH | 17A2 | FLY | | | |
| SORT | 652 | INSORT | OUTSRT | | |
| | | | | | |
| PSTFLT | 3F14 | | | | |
| RUNAC | 40B2 | RUNAC2 | STARTM | | |
| INITSK | 3E44 | ZTASK | NEWTSK | RETASK | DOWTSK |
| DOTASK | 1472 | ADDTSK | ADDTK | STPTSK | |
| ENDTSK | F14 | | | | |
| CHKWX | 518 | | | | |
| INIDEF | A4C | | | | |
| CANNIB | 125C | | | | |
| INCOMP | 70A | | | | |
| CKCRIT | 5CA | | | | |
| CKROOT | 480 | | | | |
| SCHJOB | F08 | | | | |
| SPLIT | 724 | JOIN | | | |
| GETPEO | F92 | | | | |

ENTRY POINTS

| | | | | | |
|---|---|---|---|---|---|
| PREFLT | 1550 | PRFLT | REARM | | |
| ASSIGN | 19FC | | | | |
| | | | | | |
| RECNFG | 1966 | | | | |
| UPLOAD | 1002 | | | | |
| REFUEL | A2E | | | | |
| DOWPRE | 259E | DOWPF | INWPRE | | |
| MUNEED | 271A | | | | |
| CKBILD | 1562 | | | | |
| DOBILD | 239C | DOWBLD | ENDBLD | STPBLD | |
| CKPEOP | 646 | | | | |
| CKAGE | B9A | USEAGE | RTNAGE | | |
| ADDAGE | A62 | | | | |
| CKALRT | CD2 | DOALRT | ENDALT | | |
| RELALT | 7A0 | | | | |
| | | | | | |
| ADMIN | 13FA | ZADMIN | ADMINI | ADMINO | |
| RUNSHP | A90 | | | | |
| INIREP | 3850 | ZREP | NEWREP | REREP | DOWREP |
| DOREP | CD6 | ADDREP | STPREP | | |
| ENDREP | 7E4 | | | | |
| SALVAG | 75E | | | | |
| CKCIRF | 83A | | | | |
| REPRTY | A64 | PRTY1 | PRTY2 | PRTY3 | PRTY4 |
| CKAIS | 1628 | USEAIS | AISREP | FIXAIS | |
| NRTSIT | 5CC | | | | |
| | | | | | |
| BOMB | 400E | | | | |
| REORGN | 2592 | | | | |
| REORG2 | 3060 | | | | |
| PICK | 380 | | | | |
| REBILD | 8F6 | | | | |
| INICON | D88 | | | | |
| BSEREP | CBE | ENDCE | | | |
| ENDAC | 1804 | BENDAC | | | |
| KILLAC | 652 | | | | |
| FTIME | 234 | | | | |
| SHCIRF | 680 | | | | |
| | | | | | |
| SCSHIP | DB6 | | | | |
| SHPRES | F12 | | | | |
| ORDER | 978 | | | | |
| DOSHIP | 232C | RECSUP | REFIL | | |
| STATUS | 1094 | SNDRPT | RECRPT | | |
| | | | | | |
| SHIFT | 1450 | | | | |
| REDPEO | 1730 | | | | |
| CHECK | 2DC6 | ZCHECK | | | |
| STRTSK | C1C | STTASK | REMTSK | | |
| NORRPT | C7C | RPTNOR | REDNOR | | |
| INTRUP | 6AC | ININT | OUTINT | | |
| WAIT | 6F8 | INWAIT | OUTWAT | | |
| ACWAIT | 62C | | | | |

ENTRY POINTS

| | | |
|---|---|---|
| RESET | 90E | |
| BREAK | 4FA | |
| LOSSES | 222 | |
| ACCRIT | 59E | |
| HELPER | 39A | XHELP |

Common Statements in the Simulation Segment

| | |
|---|---|
| PURGE4 | 708 |
| PURGE5 | FA0 |
| LOCAL2 | A10 |
| LOCAL3 | 1AD0 |
| LOCAL4 | 50 |

## APPENDIX E

## CALL MAP FOR TSAR SOURCE CODE

This section provides a comprehensive record of the access among the subprograms (i.e. subroutines) that constitute TSAR. It consists of a complete list of all subroutine statements, entry statements, and call statements in the order in which they appear in the TSAR source code, except that only one call to any particular subroutine (entry) is retained for each subroutine or entry statement. The statements appear as they are found in the code, complete with their argument lists.

This listing provides a ready reference for determining all calls to any given subroutine; it is especially useful when stored in a computer where a text editor can be used to locate all appearances of any specified name. (A machine readable copy will be provided along with each copy of TSAR that is distributed.)

### TAB A

```
MAIN EXECUTIVE
        CALL ENTRYP
        CALL TIMON
        CALL ERRSET(210,10,10,2,0)
        CALL INIT
        CALL INPUT
        CALL INLIST
        CALL INITIZ
        CALL TRIALS
SUBROUTINE TRIALS
        CALL ZCHECK
            IF (NEWPRT .GE. 2)          CALL STORE
            IF (ZNORS .EQ. 1)           CALL ZNOR
            IF (STATE .GT. 0)           CALL BASCAP
            IF (NEWPRT .LT. 2)          CALL STORE
        CALL NEXTR( X, TSEED)
        CALL RSTART( NXSEED )
        CALL RECALL(IT)
            IF (ZNORS .EQ. 1)           CALL ZNOR
            IF (STATE .GT. 0)           CALL BASCAP
            IF (REPORT(1,1) .GT. 0)     CALL STATUS
        CALL SCSHIP
```

```
        CALL DAYONE
        CALL MANAG
            IF (ZSHOP .EQ. 1)          CALL ZSHOPS
        CALL RESET
        CALL MANAGE(EXFLAG)
        CALL OUTPUT
SUBROUTINE MANAGE(EXFLAG)
        CALL TIMOF(TIMEU)
            IF ((TEST .GT. 0).OR.(TEST1 .LT. 0))  CALL HELPER (L, NP)
        CALL ENDCE(NP)
        CALL RUNAC(NP)
        CALL RUNSHP(NP)
        CALL ACCRIT
        CALL OUTHEP(PERIOD(1,1),20,1,N20,N0,N0,NP,TIME)
        CALL ASSETS
        CALL SHIFT
        CALL CHECK(SHOP,N0,N0,N0,N0,B,FLAG)
            IF (CREWS .EQ. 1)          CALL RELIEF
        CALL PLAN
            IF (BUILD .EQ. 1)          CALL MUNEED
        CALL EXHEAP(PERIOD(1,1),20,1,N19,N0,N0,N10)
        CALL INHEAP(PERIOD(1,1),20,1,N18,N10,LE,TIM)
        CALL CHKWX
        CALL READFT
        CALL EXHEAP(PERIOD(1,1),20,1,N19,N0,N0,N3)
        CALL INHEAP(PERIOD(1,1),20,1,N18,N3,N0,NEXTSC)
        CALL SCHFLT
        CALL OUTPUT
        CALL ADAPT
        CALL CONTRL
        CALL SCSHIP
        CALL RECSUP
        CALL DOSHIP(TIME,1)
        CALL DOSHIP(TIME,2)
            IF (AID .EQ. 0)  CALL RECRPT(LOC)
            IF (AID .EQ. 1)  CALL SNDRPT(LOC)
        CALL ADMINO
        CALL BASCAP
        CALL MODIFY(LOC, NEXT)
        CALL BOMB(LOC)
        CALL INHEAP(PERIOD(1,1),20,1,N19,NP,N0,TIME)
        CALL OUTHEP(ACN(1,18),MAXACN,1,NEH,N0,N0,NP,TT)
        CALL LAND(NP)
        CALL STARTM(NP)
        CALL RUNAC2(NP,N30026)
        CALL FLY(NP,PRTY,MISS,ACTYPE,BASE,SPARE(1,MISS,ACTYPE,BASE))
        CALL FLIGHT
        CALL ENDBLD(NP,BASE)
        CALL DOWBLD(BASE,FLAG)
        CALL DOWPF(BASE,SHOP)
        CALL REFIL(NP)
    ENTRY MANAG
```

```
        CALL INHEAP(PERIOD(1,1),20,1,NE,FE,LE,NINF)
SUBROUTINE ADAPT
        CALL NEWVAL ( DDD(1) )
SUBROUTINE CONTRL
        CALL MODHEP(REPORT(1,1),NOREPT,NORPT,I,N480)
        CALL SHPRES(N3,PART,N1,MMAXB,OBASE,MMAXB)
    ENTRY SEND(ARG1, ARG2, RETAIN)
        CALL SHPRES(N3,PART,N1,RBASE,OBASE,RBASE)
    ENTRY ORDERP(ARG1, ARG2, ARG3 )
    ENTRY GETPRT(ARG1, ARG2 )
    ENTRY FINDPT(ARG1, ARG2 )
        CALL SHPRES(N3,PART,N1,MMAXB,OBASE,MMAXB)
SUBROUTINE FERRY(NAC,ARG1,NNEXT,PFLAG)
        CALL GETPLT(ACTYPE,BASE,CREW)
        CALL FLYAC(NAC,CREW,NOW,ACTYPE,BASE)
        CALL ENDALT( NAC, ACTYPE, BASE, NO, NO )
    ENTRY LAND (NAC)
    ENTRY NEWAC(NAC)
        CALL LANDAC(NAC,LOST,BASE,NEXT)
        CALL ENDAC(NAC,ACTYPE,NEXT,1)
        CALL KILLAC(NAC, 1)
        CALL REDNOR(PART,NAC,BASE,NO,NO)
        CALL RPTNOR(PART,CRIT,NAC,SIGNAL)
        CALL OUTWAT(SHOPS(8,SHOP,BASE),SHOPS(6,SHOP,BASE),
                    SHOPS(7,SHOP,BASE),NT )
        CALL INWAIT(SHOPS(8,SHOP,NEXT),SHOPS(6,SHOP,NEXT),
                    SHOPS(7,SHOP,NEXT),TME,NN)
        CALL PSTFLT(NAC,COMBAT,DIVERT,TRANS,REPLAC,HURT)
SUBROUTINE GOHOME (SWITCH, BASE)
        CALL FERRY ( NAC, EMERG, HOME, PFLAG )
```

TAB B

```
SUBROUTINE INIT
    ENTRY STORE
    ENTRY RECALL(IT)
        CALL IRANDG( SEEDS(1) )
        CALL COMPRT( ICARGO )
            IF (DAMODE .EQ. 2)  CALL INPUTC(IT, ITEM, DAMODE, PRINT, NOATT,
                  LTATT, NOFAC, ATTACK(1,1), NITEM, DAMAGE(1,1) )
            IF (DAMODE .EQ. 3)  CALL INPUTD(IT, ITEM, DAMODE, PRINT, NOATT,
                  LTATT, NOFAC, ATTACK(1,1), NITEM, DAMAGE(1,1) )
    ENTRY DOSAVE
        CALL CRANDG(CSEEDS(1))
    ENTRY RECOVR
        CALL IRANDG(CSEEDS(1))
SUBROUTINE INPUT
        CALL ZVALUE
        CALL TESTER(I,J,D(1),NAC,AP(1),ICARGO )
            IF (SEED .GT. 0)  CALL RSTART(ISEED)
        CALL INPUTA(I,J,D,NAC,AP,ICARGO )
        CALL INPUTB( I, J, D)
            IF (UNCER .NE. 0)  CALL TTIME(AID,UNCER,AID,N0,N0,N0,N5)
        CALL INPUTE(ITRIAL,J,D,ITEM,DAMODE,PRINT,NOATT,LTATT,NOFAC,
                  ATTACK,NITEM,DAMAGE,TEST )
        CALL IRANDG( SEEDS(1) )
        CALL WRAPUP( AVGTSK(1,1), AVGREP(1,1), ICARGO  )
SUBROUTINE INPUTA(I,J,D,NAC,AP,ICARGO )
SUBROUTINE INPUTB( I, J, D)
        CALL NEWVAL( DDD(1) )
SUBROUTINE INPUTC(ITRIAL,ITEM,DAMODE,PRINT,NOATT,LTATT,NOFAC,
                  ATTACK,NITEM,DAMAGE )
    ENTRY INPUTD(ITRIAL, ITEM,DAMODE,PRINT,NOATT,LTATT,NOFAC,
                  ATTACK,NITEM,DAMAGE )
    ENTRY INPUTE(ITRIAL,ARG1,E,ITEM,DAMODE,PRINT,NOATT,LTATT,NOFAC,
                  ATTACK,NITEM,DAMAGE,ARG2 )
        CALL INHEAP(ATTACK(1,1),LTATT,1,NOATT,FE,LE,ATTIME)
SUBROUTINE WRAPUP( AVGTSK, AVGREP, ICARGO )
        CALL ICHECK
        CALL COMPRT( ICARGO )
        CALL AVGTME( AVGTSK(1,1), AVGREP(1,1) )
        CALL RREQTS
SUBROUTINE ICHECK
        CALL CKRQT2(NOPART,NTYPE,JOBCON,MNTLMT,TEST)
        CALL ZSPLIT( 0 )
        CALL CKNET( AC, TASK, PART, PTASK, NR, LEVEL, VERIFY, FLAG )
        CALL CKDAM(AC,TASK,PART,PTASK,NR,LEVEL,VERIFY,FLAG,BATTLE)
        CALL CKRQMT
        CALL PSHORT(AID1,AID1,SHORT,RANDM)
        CALL IHELPC
        CALL HELPCK(N,TASK,SHOP,PEOP,1,CLASS)
SUBROUTINE HELPCK(N,ITEM,SHOP,PEOP,TYPE,CLASS)
    ENTRY IHELPC
```

```
SUBROUTINE INLIST
        CALL HEADER
        CALL LIST1(NOTSK,HTH(M),R(M),NAME(1,M),NAME(2,M),TSKRQT(1,1),0)
        CALL LIST2(NOPEOP,HTH(M),R(M),NAME(1,M),NAME(2,M),PEOPLE(1,1,1),BMAX)
        CALL LIST3(20,MAXM,MAXT,NAME(1,M),NAME(2,M),ACMDTA(1,1,1))
        CALL LIST4(NOBILD,HTH(M),R(M),NAME(1,M),NAME(2,M),MUNRQT(1,1),0)
        CALL LIST5(NOFA,HTH(M),R(M),NAME(1,M),NAME(2,M),FACLTY(1,1,1),BMAX)
SUBROUTINE HEADER
        CALL DATIME(3,LOC)
SUBROUTINE INITIZ
        CALL TIMOF(TIMEU)
SUBROUTINE ZSHOPS
        CALL SHPRQ(ND, NT, N1, TASK )
        CALL ZTASK(SHOPS(1,1,BASE),PEOPLE(1,1,BASE),PARTS(1,1,1),
                   AGESTK(1,1,BASE), MUNSTK(1,1,BASE), TRAP(1,BASE),
                   NT, TASK, RTASK, START, WHEN )
        CALL ZADMIN(UNIT,RTASK,BASE,BASE,NUM)
        CALL ZREP(SHOPS(1,1,BASE),PEOPLE(1,3,BASE),AGESTK(1,2,BASE),
                   PARTS(1,1,1), UNIT, BASE, OBASE, WHEN )
SUBROUTINE TESTER(I,J,D,NAC,AP,ICARGO )
        CALL INPUTA(I,J,D(1),NAC,AP(1),ICARGO )
        CALL INPUTB(I,J,D(1))
SUBROUTINE MODIFY ( NF, NEXT )
        CALL OUTHEP( CHANGE(1,1), NCHG, 0, NEV, FEV, LEV, NF, TIME )
    ENTRY NEWVAL ( DDD )
        CALL INHEAP( CHANGE(1,1), NCHG, 0, NEV, FEV, LEV, TIME )
        CALL MODHEP(PERIOD(1,1),20,N20,N14,ADD)
    ENTRY ZVALUE
SUBROUTINE CKNET(AC, TASK, PART, PTASK, NR, LEVEL, VERIFY,FLAG)
    ENTRY CKDAM(AC,TASK,PART,PTASK,NR,LEVEL,VERIFY,FLAG,ARG1 )
        CALL CKRQT( AC, ARG1, TASK, PTASK, RQMNT, TOTIME, BATTLE, NSPLIT)
            IF (TSKRQT(TASK,15) .NE. 0)  CALL CKSPLT(TASK, NSPLIT)
SUBROUTINE CKRQT( ACTYPE, PART, TASK, PTASK, RQMNT, TOTIME,
                   BATTLE, NSPLIT )
        CALL TEMPRQ(PART,ACTYPE,NSPLIT,PROB)
    ENTRY CKRQMT
    ENTRY CKRQT2( ARG2, ARG3, ARG4, ARG5, ARG6 )
SUBROUTINE CKSPLT(TASK, NSPLIT )
    ENTRY TEMPRQ(ARG1, ARG2, NSPLIT, PROB)
    ENTRY ZSPLIT(SWITCH)
SUBROUTINE ZNOR
        CALL CKNET(AC, TASK, NPART, PROB, NR, LEVEL, VERIFY, FLAG )
        CALL STTASK(NF, TASK, TYPE, B)
        CALL RPTNOR(PART, CRIT, NF, B)
```

TAB C

```
SUBROUTINE COMPRT ( ICARGO )
        CALL IPARTS( ICARGO )
        CALL TTIME(PT,UNCER,PA,NO,NO,NO,N5)
SUBROUTINE IPARTS(ICARGO )
        CALL CKNET( AC, TASK, PART, PTASK, NR, LEVEL,VERIFY, FLAG )
        CALL CKNRTS( TBASE, PART, POST, WOST, CBRT, CWBRT, CPOST, CWOST,
            REM1, CDMN, PFLAG, SFLAG, CIRF, ONRTS, CNRTS, NRTS2,
            T1, T4, TRP, TRW, ONBASE, INCIRF )
        CALL PSHORT(NS, TS, SHORT(ITEM), RANDM )
        CALL IPART2 ( CIRF, ICARGO, ZAP )
SUBROUTINE IPART2( CIRF, ICARGO, ZAP )
SUBROUTINE CKNRTS(BASE,PART,POST,WOST,CBRT,CWBRT,CPOST,CWOST,
            REM1, CDMN, PFLAG, SFLAG, CIRF, ONRTS, CNRTS, NRTS2,
            T1, T4, TRP, TRW, ATBASE, ATCIRF )
SUBROUTINE RREQTS
            IF (SHOP .EQ. 26)   CALL REQTS3( ACTYPE,MMAXM )
        CALL REQTS1(PROB,CRIT(1),TASK,PART,BASE,ACTYPE,MMAXM )
        CALL CKNET( ACTYPE, TASK, PART, PTASK, NR, LEVEL, VERIFY, FLAG)
        CALL REQTS2( PROB, PART, TASK, CRIT(1)  )
SUBROUTINE REQTS1(PROB,CRIT,TASK,PART,ARG1,ARG2,ARG3)
    ENTRY REQTS2 (PROB, PART, TASK, CRIT )
    ENTRY REQTS3 ( ARG2, ARG3 )
SUBROUTINE OUTPUT
            IF (VBREAK .EQ. 1)  CALL BREAK
        CALL TIMOF(TIMEU)
        CALL JOBLST( RECORD(1,1,1), MMAXRC, NNSCRL )
            IF (AID .EQ. 0)  CALL DELAY2
        CALL DELAY1
SUBROUTINE ASSETS
SUBROUTINE ASSET2(BASE)
SUBROUTINE TIMES
    ENTRY DOTIME(KIND, SHOP, BASE, TIME)
    ENTRY WAITS(AID, BASE, TIME)
SUBROUTINE DELAYS
    ENTRY DELAY1
        CALL WAITS(WAITSK(NF,10), BASE, TIME)
    ENTRY DELAY2
        CALL ASSET2(BASE)
SUBROUTINE PSHORT(N, T, SHORT, RANDM )
SUBROUTINE JOBLST( RECORD, MMAXRC, NNSCRL )
SUBROUTINE DEFERS
```

TAB D

```
SUBROUTINE READFT
    ENTRY DAYONE
        CALL INSORT(TIME)
    ENTRY SCHFLT
        CALL INSORT(TIME)
            IF (SELECT .GT. 0)  CALL FRAG(SINGLE,DOFRAG,DUMMY)
    ENTRY SORTIE( ARG1, DATA, ARG2 )
        CALL INSORT(TIME)
        CALL FRAG(N, DOFRAG, BASE)
        CALL MODHEP(PERIOD(1,1),15,NE,MC,SUB)
SUBROUTINE FRAG(SINGLE, DOFRAG, BASE)
        CALL OUTSRT(NF)
        CALL INSORT(TIME)
SUBROUTINE PLAN
        CALL FERRY( NF, BASE, NXBASE, PFLAG )
        CALL PLAN1(SUPPLY(1,MISS,ACTYPE),NF,ALRT2,ALRT4,
                   BASE,ACTYPE,MISS,SORDEF(1,1,MISS,ACTYPE,BASE))
        CALL REASSG
        CALL INIDEF(NF,WORKTM,BASE,FLAG)
SUBROUTINE PLAN1(SUPPLY,NF,ALERT1,ALERT2,BASE,ACTYPE,MISS,SORDEF)
SUBROUTINE BASCAP
        CALL FERRY( NF, BASE, NNEXT, PFLAG )
SUBROUTINE REASSG
        CALL ENDALT( NA, ACTYPE, BASE, NO, NO)
        CALL REMTSK(NF,NP,2,TASK,BASE)
        CALL STTASK(NF,NTASK,1,BASE)
            IF ((FLAG .EQ. 6).AND.(ACN(NF,7) .NE. 0))  CALL STARTM(NF)
        CALL REARM(NF,N27,BASE)
        CALL INHEAP(RESUPP(1,3),LLGQ,0,NEG,FEG,LEG,ARRIVE)
SUBROUTINE FLYERS
    ENTRY GETPLT(ARG1, ARG2, TEMP)
    ENTRY SAVPLT(ARG1, ARG2, TEMP)
    ENTRY FLYAC( NAC, ARG1, TIME, ARG2, ARG3)
    ENTRY LANDAC(NAC, LOST, ARG1, NB )
    ENTRY RELIEF
SUBROUTINE DISABL(PCTLOS, B)
            IF (REFILL(1,CLASS).GE.100) CALL ORDER(B,CLASS,A,N)
SUBROUTINE FLIGHT
        CALL GETPLT(ACTYPE,BASE,CREW)
        CALL SAVPLT(ACTYPE,BASE,CREW)
        CALL OUTSRT(FLT)
        CALL LAUNCH(FLT,M,ML,PRY,PRTY,NTPY,ACA(1,MISS,ACTYPE,BASE),
                   ALERT(1,MISS,ACTYPE,BASE),SPARE(1,MISS,ACTYPE,BASE),
                   MISS,ACTYPE,BASE)
            IF (AIDALT(ACTYPE) .EQ. 1)  CALL ENDALT(NA,ACTYPE,BASE,NO,NO)
        CALL CKALRT(NS,ACTYPE,BASE,START)
            IF(AIDALT(ACTYPE).EQ.1)CAL DOALRT
SUBROUTINE LAUNCH(FLT,M,ML,PRY,PRTY,NTPY,ACA,ALERT,SPARE,
                   MISS,ACTYPE,BASE )
            IF (AIDALT(ACTYPE) .EQ. 1)  CALL ENDALT(NA,ACTYPE,BASE,NO,NO)
```

```
          CALL INHEAP(ACN(1,18),MAXACN,1,NEH,NA,0,TIMOUT)
      ENTRY FLY(NA,PRTY,MISS,ACTYPE,BASE,SPARE)
          CALL ENDALT(NA, ACTYPE, BASE, NO, NO )
          CALL TTIME(TIM,TD,TIME,NO,NO,NO,NO)
          CALL FLYAC(NA,CREW,TIME,ACTYPE,BASE )
          CALL INHEAP(ACN(1,18),MAXACN,1,NEH,NA,NO,TIME)
SUBROUTINE SORT
      ENTRY INSORT(TIME)
      ENTRY OUTSRT(FLT)
```

TAB E

```
      SUBROUTINE PSTFLT(NAC,COMBAT,DIVERT,TRANS,REPLAC,HURT)
            CALL TTIME(LDELAY,TD,T,NO,NO,NO,NO)
            CALL INHEAP(ACN(1,18),MAXACN,1,NEH,NAC,NO,T)
            CALL CKCRIT(NTASK,CKPART,DODEF,NOWRQD,MMAX,SUM,BASE)
            CALL REMTSK( NAC, NP, 2, NTASK, BASE)
            CALL CKCRIT(NTASK,CKPART,NO,NO,MMAX,SUM,BASE)
            CALL SHPRQ( NP, NAC, ACTYPE, TRANS, ITEM )
            CALL SHPRQT(NAC,NRN,SHOP,ACTYPE,TRANS,TSKP,TASK,ND)
            CALL SCHJOB(NAC,ACTYPE,BASE,BT,NJOB,TASKS(1),TCRIT(1),BD(1,1) )
            CALL STTASK(NAC, BD(NNN,1), 2, BASE)
            CALL STTASK(NAC, TASK, 1, BASE)
            CALL REMTSK(NAC,NP,2,TASK,BASE)
            CALL STTASK(NAC,NTASK,1,BASE)
            CALL INHEAP(RESUPP(1,3),LLGQ,0,NEG,FEG,LEG,ARRIVE)
      SUBROUTINE RUNAC(NP)
            CALL DOTIME(1, SHOP, BASE, TSKTME)
            CALL ENDTSK(NP,SHOP,SHOPS(1,SHOP,BASE),PEOPLE(1,1,BASE),
                  AGESTK(1,2,BASE),NAC,BASE )
            CALL CHECK(NO,NO,NO,PEOP,NO,BASE,BFLAG)
          ENTRY RUNAC2( ARG1, ARG2 )
            CALL PREFLT(TSK,NAC)
            CALL JOIN(TASK,NAC,NOTYET)
            CALL CKROOT(TASK,NAC,CANCL)
            CALL SPLIT(NAC, TASK, JOINT, SKIP)
            CALL NEWTSK(SHOPS(1,1,BASE),PEOPLE(1,1,BASE),PARTS(1,1,1),
                  AGESTK(1,1,BASE), MUNSTK(1,1,BASE), TRAP(1,BASE),
                  NAC, TASK, RTASK, START, BEGUN )
            CALL SPLIT(NAC, TASK, JOINT, SKIP )
            CALL DOWTSK(SHOPS(1,1,BASE),PEOPLE(1,1,BASE),PARTS(1,1,1),
                  AGESTK(1,1,BASE),MUNSTK(1,1,BASE),TRAP(1,BASE),NF,START)
            CALL INWPRE(NAC,SHOP,BASE,NF)
                IF (TASK.EQ.30029)  CALL INWPRE(NAC,SHOP,BASE,NF)
            CALL DOWPRE(NAC,TASK)
            CALL INWPRE(NAC,N26,BASE,NF)
            CALL REMTSK(NAC,ACN(NAC,7),1,NO,NO)
            CALL INHEAP(RESUPP(1,3),LLGQ,0,NEG,FEG,LEG,ARRIVE)
            CALL ENDAC(NAC,ACTYPE,BASE,2)
            CALL DISABL(NN1, BASE)
            CALL FERRY( NAC, BASE, NXBASE, PFLAG)
            CALL INIDEF(NAC,WORKTM,BASE,SFLAG)
            CALL PRFLT(TASK,NAC)
            CALL TTIME(TIME,TD,DELAY,NO,NO,NO,NO)
            CALL INHEAP(ACN(1,18),MAXACN,1,NEH,NAC,NO,DELAY)
          ENTRY STARTM( ARG1 )
            CALL REMTSK(NAC,NT,1,NO,NO)
            CALL GOHOME ( NAC, BASE )
            CALL PRFLT(TASK,NAC)
            CALL NEWTSK(SHOPS(1,1,BASE),PEOPLE(1,1,BASE),PARTS(1,1,1),
                  AGESTK(1,1,BASE), MUNSTK(1,1,BASE), TRAP(1,BASE),
                  NAC, TASK, RTASK, START, BEGUN )
```

```
SUBROUTINE INITSK
    ENTRY ZTASK(SHOPS,PEOPLE,PARTS,AGESTK,MUNSTK,TRAP,
                ARG1, ARG2, ARG3, START, BEGUN )
    ENTRY NEWTSK(SHOPS,PEOPLE,PARTS,AGESTK,MUNSTK,TRAP,
                ARG1, ARG2, ARG3, START, BEGUN )
    ENTRY RETASK(SHOPS,PEOPLE,AGESTK,ARG4,START )
    ENTRY DOWTSK(SHOPS,PEOPLE,PARTS,AGESTK,MUNSTK,TRAP,ARG4,START)
        CALL INCOMP(INC,ACN(NAC,9),ACN(NAC,5),NWAIT)
        CALL CKAGE(NAC,SQD,ACTYPE,AGESTK(1,2),AGE,AGE1,AGE2,
                A1, A2, K, WHY, AVAIL )
        CALL GETPEO(BASE,NAC,SQD,TASK,NALT, ENOUGH, MFLAG, PFLAG, WAITT,
             WHY,SHOPS(1,SHOP),PEOPLE(1,3),AGESTK(1,2),PEO(1),PEOP(1),
             NUM(1),XPEO(1),XPEOP(1),XNUM(1) )
        CALL CANNIB(NAC,NTASK,PART, CFLAG)
        CALL RPTNOR(PART,CRIT,NAC,BASE)
            IF ((TAID.EQ.2).OR.(TAID.GE.4))CALL REDNOR(PART,NAC,BASE,NO,NO)
        CALL TTIME(TIM,TD,MTIME,HURRY1,REDUC1,SAVE1,NO)
        CALL USEAGE(NAC, AGESTK, AGE1, AGE2, A1, A2, K )
        CALL ADDTK(SHOPS(1,SHOP),NAC,NTASK,RTASK,NALT,PEO(1),AGE,
                ENDTME, WHEN, XPEO(1) )
        CALL ACWAIT(SHOPS(1,SHOP),NO,NTASK,NAC,PART,NO,NO,NO,
                WHY,WHEN,NOW,RTASK )
        CALL OUTINT(SHOPS(5,SHOP),SHOPS(3,SHOP),SHOPS(4,SHOP),TASKP)
        CALL OUTWAT(SHOPS(8,SHOP),SHOPS(6,SHOP),SHOPS(7,SHOP),TASKP)
        CALL WAITS(WHY, BASE, DELAY)
        CALL ORDER(DESTIN,CLASS,PART,N1)
        CALL ADMINI(PART,TASK,BASE,BASE,AVPT)
        CALL SHPRES(N5,-PART,N1,BASE,NXBASE,BASE)
SUBROUTINE DOTASK
    ENTRY ADDTSK(SHOPS, ARG1, TASK, RTASK, NALT, PEORES, AGERES,
                TME, WHEN,YXPEO )
    ENTRY ADDTK(SHOPS, ARG1, TASK, RTASK, NALT, TEAMS, AGERES,
                TME, WHEN, XTEAMS )
        CALL INHEAP(TASKQ(1,4),LLTQ,0,NET,FET,LET,TIME)
    ENTRY STPTSK(SHOPS,PEOPLE,AGESTK,IT,CFLAG)
        CALL ININT(SHOPS(5),SHOPS(3),SHOPS(4),TIME,NLET)
        CALL EXHEAP(TASKQ(1,4),LLTQ,0,NET,FET,LET,IT)
        CALL RTNAGE(NAC, SQD, TYPE, AGESTK(1), AGE )
SUBROUTINE ENDTSK(NP,SHOP,SHOPS,PEOPLE,AGESTK,NAC,BASE)
        CALL OUTHEP(TASKQ(1,4),LLTQ,0,NET,FET,LET,NP,TIME)
        CALL RTNAGE(NAC, SQD, TYPE, AGESTK(1), AGE)
        CALL DOWPRE(NAC,TASK)
        CALL CHECK(NO, NO, NO, PEOP1, NO, BASE, NO)
SUBROUTINE CHKWX
        CALL INIDEF(NF, MAXTWX, BASE, SFLAG)
SUBROUTINE INIDEF (NAC,WORKTM,BASE,SFLAG)
        CALL NEWTSK(SHOPS(1,1,BASE),PEOPLE(1,1,BASE),PARTS(1,1,1),
                AGESTK(1,1,BASE), MUNSTK(1,1,BASE), TRAP(1,BASE),
                NAC, NTASK, RTASK, START, BEGUN )
        CALL REMTSK(NAC,DT,2,TASK,BASE)
SUBROUTINE CANNIB(NAC,RTASK,PART,CFLAG)
        CALL STTASK(NA, AID, 2, BASE)
```

```
        CALL RPTNOR(PART,CRIT,NA,BASE)
SUBROUTINE INCOMP(INC,ITP,IWP,NWAIT)
SUBROUTINE CKCRIT(TASK,CKPART,DODEF,NOWRQD,MMAX,SUM,BASE)
SUBROUTINE CKROOT(TASK, NAC, CANCEL)
SUBROUTINE SCHJOB(NAC,ACTYPE,BASE,BT,NJOB,TASKS,TCRIT,BD)
        CALL REMTSK(NAC, NP, 1, NO, NO)
        CALL STTASK(NAC, TASK, 1, BASE )
SUBROUTINE SPLIT(NAC,TASK,JOINT,SKIP)
    ENTRY JOIN(TASK, NAC, NOTYET)
SUBROUTINE GETPEO( BASE, NAC, SQD, TASK, NALT, ENOUGH, MFLAG,
                   PFLAG, WAITT, WHY, SHOPS, PEOPLE, AGESTK,
                   PEO, PEOP, NUM, XPEO, XPEOP, XNUM  )
        CALL CKPEOP( BASE, KIND, PEO(I), PEOP(I), NUM(I), XPEO(I),
           XPEOP(I), XNUM(I), AVPEO(I), PFLAG, SFLAG(I), PEOPLE(I), SQD)
        CALL STPREP(SHOPS(1),PEOPLE(1),AGESTK(1),NR,BASE)
        CALL STPTSK(SHOPS(1), PEOPLE(1), AGESTK(1), NR, CFLAG )
```

TAB F

```
SUBROUTINE PREFLT(TASK,NAC)
        CALL DOWPRE(NAC,TASK)
    ENTRY PREFLT(TASK,NAC)
        CALL ASSIGN(NAC, ACA(1,1,ACTYPE,BASE),ALERT(1,1,ACTYPE,BASE),
                  PTZ(1,ACTYPE,BASE),SPARE(1,1,ACTYPE,BASE),
                  SORDEF(1,1,1,ACTYPE,BASE), MISS)
        CALL INCOMP(NP, ACN(NAC,9), ACN(NAC,5), NWAIT)
        CALL TTIME(TIME,TD,T,NO,NO,NO,NO)
        CALL INHEAP(ACN(1,18),MAXACN,1,NEH,NAC,0,T)
    ENTRY REARM (NAC, TASK, BASE)
        CALL RECNFG(NAC,SHOPS(1,1,BASE),PEOPLE(1,3,BASE),
                  AGESTK(1,2,BASE),MUNSTK(1,1,BASE),TRAP(1,BASE),WHEN,WHY)
        CALL UPLOAD(NAC,SHOPS(1,28,BASE),PEOPLE(1,3,BASE),
                  AGESTK(1,2,BASE),WHEN,WHY)
        CALL REFUEL(NAC,SHOPS(1,29,BASE),PEOPLE(1,3,BASE),
                  AGESTK(1,2,BASE),POLSTK(BASE),NT,WHEN,WHY)
        CALL ACWAIT(SHOPS(1,SHOP,BASE),FLAG1,TASK,NAC,FLAG2,
                  AGE,PEO,TIME,WHY,NOW,NOW,NO)
SUBROUTINE ASSIGN(NAC,ACA,ALERT,PTZ,SPARE,SORDEF, MISS)
        CALL CKALRT( NAC, ACTYPE, BASE, START )
        CALL DOALRT
SUBROUTINE RECNFG(NAC,SHOPS,PEOPLE,AGESTK,MUNSTK,TRAP,WHEN,WHY)
        CALL CKPEOP(BASE, KIND1, AID2, PEOP1, NPEO1, XPEO1, XPEOP1, XNUM1,
                  AVPEO1, PFLAG1, SFLAG, PEOPLE(1),SQD  )
        CALL CKAGE(NAC, SQD, TYPE, AGESTK(1), AID1, EQP1, EQP2,
                  A1, A2, K1, WHY, AVAIL)
        CALL USEAGE(NAC, AGESTK, EQP1, EQP2, A1, A2, K1 )
        CALL TTIME (TIM, TD, UP1, HURRY2, REDUC2, SAVE2, NO )
        CALL ADDTSK( SHOPS(1,27), NAC, N30027, NO,NO,AID2,AID1,
                  TIME1, WHEN, XPEO1 )
SUBROUTINE UPLOAD (NAC,  SHOPS,PEOPLE,AGESTK, WHEN,WHY )
        CALL TTIME(TIME,TD,T1,HURRY2,REDUC2,SAVE2,NO)
        CALL CKPEOP(BASE, KIND1, AID2,PEOP1,NPEO1, XPEO1, XPEOP1,XNUM1,
                  AVPEO1, PFLAG1, SFLAG, PEOPLE(1), SQD )
        CALL CKAGE(NAC, SQD, TYPE, AGESTK(1), AID1, EQP1, EQP2,
                  A1, A2, K1, WHY, AVAIL )
        CALL ADDTSK(SHOPS(1), NAC, N30028,NO,NO,AID2,AID1,T1,WHEN,XPEO1)
        CALL USEAGE(NAC, AGESTK, EQP1, EQP2, A1, A2, K1 )
        CALL TTIME(TIME,TD,T2,HURRY2,REDUC2,SAVE2,NO)
SUBROUTINE REFUEL(NAC,SHOPS,PEOPLE,AGESTK,POLSTK,NT,WHEN,WHY)
        CALL CKAGE(NAC, SQD, ACTYPE, AGESTK(1), AGE, AGE1, AGE2,
                  A1, A2, K, WHY, AVAIL )
        CALL USEAGE(NAC, AGESTK, AGE1, AGE2, A1, A2, K )
        CALL TTIME(TIME,TD,T,HURRY2,REDUC2,SAVE2,NO)
        CALL ADDTSK(SHOPS(1),NAC,N30029,NO,NO,PEO,AGE,TIME,WHEN,NO)
SUBROUTINE DOWPRE( ARG1, ARG2 )
    ENTRY DOWPF( ARG3, ARG4 )
        CALL RECNFG(NA,SHOPS(1,1,BASE),PEOPLE(1,3,BASE),
             AGESTK(1,2,BASE),MUNSTK(1,1,BASE),TRAP(1,BASE),WHEN,WHY)
        CALL CKPEOP( BASE, KIND, PEO1,PEOP1,NUM1, XPEO,XPEOP,XNUM,
```

```
                AVPEO, PFLAG, SFLAG, PEOPLE(1,3,BASE), SQD )
        CALL CKAGE(NA, SQD, TYPE, AGESTK(1,2,BASE), AGE1, AGE3, AGE4,
                A3, A4, K, WHY, AVAIL )
        CALL ADDTSK(SHOPS(1,SHOP,BASE),NA,N30027,NO,NO,PEO1,AGE1,
           TIME, WHEN, XPEO )
        CALL USEAGE(NA, AGESTK, AGE3, AGE4, A3, A4, K )
        CALL UPLOAD(NA,SHOPS(1,SHOP,BASE),PEOPLE(1,3,BASE),
                AGESTK(1,2,BASE), WHEN, WHY )
    ENTRY INWPRE( ARG1, ARG2, ARG3, ARG4 )
        CALL INCOMP(NI, ACN(NA,9), ACN(NA,5), NWAIT)
        CALL TTIME(TIME,TD,T,NO,NO,NO,NO)
        CALL INHEAP(ACN(1,18),MAXACN,1,NEH,NA,0,T)
        CALL REFUEL(NA,SHOPS(1,29,BASE),PEOPLE(1,3,BASE),
                AGESTK(1,2,BASE), POLSTK(BASE), NT, WHEN, WHY )
        CALL WAITS(WAITSK(NP,10), BASE, TIME)
        CALL PRFLT(NXTASK,NA)
SUBROUTINE MUNEED
        CALL CKBILD(DEMAND(1,1,1,1), PRY(1,1), MAXWT(1), B)
        CALL DOBILD(AMMO,BUSY,N10,MAXWT(1),B)
SUBROUTINE CKBILD(DEMAND, PRY, MAXWT, B)
        CALL DOBILD(AMMO,BUSY(AMMO),T,MAXWT(1),B)
SUBROUTINE DOBILD(ARG1,ARG2,ARG3,MAXWT,BASE)
    ENTRY DOWBLD(BASE,FLAG)
        CALL TTIME(TIM,TD,TIME,HURRY4,REDUC4,SAVE4,NO)
        CALL CKPEOP( BASE, KIND, PEO,PEOP,NUMP, XPEO,XPEOP,XNUM,
                AVPEO, NO, SFLAG, PEOPLE(1,3,BASE), NO  )
        CALL INHEAP(BUILDQ(1,2),LLBQ,0,NEB,FEB,LEB,ENDTME)
        CALL WAITS(WHY,BACKLG(5,JOBP),TIME)
    ENTRY ENDBLD(NP,BASE)
        CALL OUTHEP(BUILDQ(1,2),LLBQ,0,NEB,FEB,LEB,PO,TIME)
        CALL ADMINI(UNIT, NO, BASE, BASE, NO)
        CALL DOTIME(2,SHOP,BASE,TIME)
    ENTRY STPBLD(NP,BASE)
        CALL EXHEAP(BUILDQ(1,2),LLBQ,0,NEB,FEB,LEB,NP)
SUBROUTINE CKPEOP ( BASE, KIND, PEO, PEOP, NUM,
                XPEO, XPEOP, XNUM, AVPEO, PFLAG, SFLAG, PEOPLE, SQD )
SUBROUTINE CKAGE(NAC, SQD, ACTYPE, AGESTK, AGE, ARG1, ARG2,
                A1, A2, K, WHY, AVAIL  )
    ENTRY USEAGE(NAC, AGESTK, ARG1, ARG2, A1, A2, K)
    ENTRY RTNAGE(NAC, SQD, TYPE, AGESTK, AGE )
        CALL ADMINI(UNIT, NO, BASE, BASE, NO)
SUBROUTINE ADDAGE ( AGE, ADD, BASE)
SUBROUTINE CKALRT( NAC, ACTYPE, ARG1, ARG2 )
    ENTRY DOALRT
    ENTRY ENDALT( NAC, ACTYPE, ARG1, PEOLOS, AGELOS)
        CALL CHECK(NO, NO, NO, PEOP, NO, BASE, NO)
            IF (REFILL(1,2) .GE. 100)  CALL ORDER(BASE,N2,AGE1,N1)
        CALL ADMINI(UNIT, NO, BASE, BASE, NO)
SUBROUTINE RELALT( BASE, MOD )
        CALL ENDALT(NF, I, BASE, NO, NO )
```

TAB G

```
SUBROUTINE ADMIN
    ENTRY ZADMIN( ARG1, ARG2, ARG3, ARG4, ARG5 )
    ENTRY ADMINI( ARG1, ARG2, ARG3, ARG4, ARG5 )
        CALL SHPRES(N3,-UNIT,N1,BASE,MAXB,BASE)
        CALL TTIME(DELY,TD,TIME,NO,NO,NO,NO)
        CALL INHEAP(LIMBO(1,4),NNLIMB,0,NEX,FEX,LEX,TIME)
        CALL NEWREP(SHOPS(1,1,BASE),PEOPLE(1,3,BASE),AGESTK(1,2,BASE),
                PARTS(1,1,1), UNIT, BASE, OBASE, WHEN, FIXSRU, AVPT )
    ENTRY ADMINO
        CALL FIXAIS( AIS, WHEN, BASE )
        CALL CHECK(NO,NO,NO,NO,AGE,BASE,NO)
        CALL OUTHEP(LIMBO(1,4),NNLIMB,0,NEX,FEX,LEX,NF,TIME)
        CALL NEWREP(SHOPS(1,1,BASE),PEOPLE(1,3,BASE),AGESTK(1,2,BASE),
                PARTS(1,1,1), UNIT, BASE, OBASE, WHEN, FIXSRU, AVPT )
SUBROUTINE RUNSHP(NP)
        CALL SEND(PART,BASE,RETAIN)
        CALL SHPRES(NN,UNIT,N1,BASE,OBASE,OBASE)
        CALL ENDREP(NP,SHOP,SHOPS(1,SHOP,BASE),PEOPLE(1,1,BASE),
                AGESTK(1,2,BASE), PART, BASE  )
        CALL DOTIME(SET, SHOP, BASE, TIME)
        CALL CHECK(NO,NO,NO,PEOP,NO,BASE,NO)
        CALL CKCIRF(SHOP,REPAIR,BASE)
SUBROUTINE INIREP
    ENTRY ZREP(SHOPS,PEOPLE,AGESTK,PARTS,ARG1,BASE,ARG2,ARG3)
    ENTRY NEWREP(SHOPS,PEOPLE,AGESTK,PARTS,ARG1,BASE,ARG2,
                ARG3, ARG4, ARG5 )
    ENTRY REREP(SHOPS,PEOPLE,AGESTK,PARTS,ARG6,BASE)
    ENTRY DOWREP(SHOPS,PEOPLE,AGESTK,PARTS,ARG6,BASE)
        CALL ADMINI(FIXSRU,NO,BASE,BASE,AVSRU)
        CALL ORDERP(SRU,BASE,FLAG)
        CALL SALVAG(SRU,LRU,SHOP,BASE,SHOPS(1,1),PARTS(1,1,BASE),FLAG)
        CALL ORDER(BASE,CLASS,SRU,N1)
        CALL CKAIS(BASE,HELP,AGE1,AIS,EXTRA,WHY)
        CALL TTIME(TIM,TD,TIME,HURRY3,REDUC3,SAVE3,NO)
        CALL USEAIS
        CALL ADDREP(SHOPS(1,SHOP),REPAIR,NALT,LRU,PEO,AGE,ENDTME,
                BASE,OBASE,WHEN)
        CALL PRTY4(PART,IMPORT)
        CALL INWAIT(SHOPS(16,SHOP),SHOPS(14,SHOP),SHOPS(15,SHOP),
                TIME, NLEW  )
        CALL PRTY3(PART,NORS,TIME)
        CALL OUTINT(SHOPS(13,SHOP),SHOPS(11,SHOP),SHOPS(12,SHOP),JOBP)
        CALL OUTWAT(SHOPS(16,SHOP),SHOPS(14,SHOP),SHOPS(15,SHOP),JOBP)
        CALL WAITS(AID, BASE, TIME)
        CALL SHPRES(N3,-ITEM,N1,BASE,NXBASE,BASE)
SUBROUTINE DOREP
    ENTRY ADDREP(SHOPS,REPAIR,NALT,LRU,PEORES,AGERES,TIME,
                BASE, OBASE, WHEN)
        CALL INHEAP(REPQ(1,4),LLRQ,0,NER,FER,LER,TIME)
    ENTRY STPREP(SHOPS,PEOPLE,AGESTK, IT ,BASE)
```

```
        CALL ININT(SHOPS(13),SHOPS(11),SHOPS(12),TIME,NLEI)
        CALL EXHEAP(REPQ(1,4),LLRQ,0,NER,FER,LER,IT)
        CALL AISREP( AIS, PART, NO, -1, BASE )
SUBROUTINE ENDREP(NP,SHOP,SHOPS,PEOPLE,AGESTK,PART,BASE)
        CALL OUTHEP(REPQ(1,4),LLRQ,0,NER, FER, LER, PO, TIME)
        CALL AISREP( AIS, PART, NO, 0, BASE )
        CALL ADMINI(UNIT, NO, BASE, BASE, NO )
SUBROUTINE SALVAG(SRU,LRU,SHOP,BASE,SHOPS,PARTS,FLAG)
            IF (REFILL(1,CLASS) .GE. 100)  CALL ORDER(DESTIN,CLASS,LRU,N1)
        CALL OUTWAT(SHOPS(16,SHOP),SHOPS(14,SHOP),SHOPS(15,SHOP),NF)
SUBROUTINE CKCIRF(SHOP,REPAIR,BASE)
        CALL REREP(SHOPS(1,1,BASE), PEOPLE(1,3,BASE), AGESTK(1,2,BASE),
                   PARTS(1,1,1),NF, BASE)
        CALL PRTY2(PART,NORS,DEMAND)
        CALL PRTY1(PART,NORS,DEMAND)
        CALL DOWREP(SHOPS(1,1,BASE), PEOPLE(1,3,BASE), AGESTK(1,2,BASE),
                   PARTS(1,1,1), NF, BASE)
SUBROUTINE REPRTY
    ENTRY PRTY1(PART,NORS,ARG1)
    ENTRY PRTY2(PART,NORS,ARG1)
    ENTRY PRTY3(PART,NORS,TIME)
    ENTRY PRTY4(PART,ARG1)
SUBROUTINE CKAIS( ARG1, ARG2, AGE, ARG3, EXTRA, WHY)
    ENTRY USEAIS
    ENTRY AISREP( ARG3, ARG2, ARG4, LOST, ARG1 )
        CALL TTIME( TME, TD, TIM, HURRY(BASE,3,2),REDUCE(BASE,3,2),
                   SAVE(BASE,3,2), NO )
        CALL INHEAP(LIMBO(1,4), NNLIMB,0,NEX,FEX,LEX,TIM)
        CALL EXHEAP(LIMBO(1,4),NNLIMB,0,NEX,FEX,LEX,I)
    ENTRY FIXAIS( ARG3, NT, ARG1 )
```

TAB H

```
SUBROUTINE BOMB(LOC)
        CALL LOSSES(AID, PCTLOS, LOST, NONUNI)
        CALL DISABL(PCTLOS,BASE)
        CALL AISREP(AIS, NO, STRING, 1, BASE)
        CALL LOSSES( STOCK, PCTLOS, LOST, NONUNI)
        CALL ORDER(BASE,CLASS,NAGE,TOT)
        CALL ENDALT(NA, ACTYPE, BASE, PEOLOS, AGELOS )
        CALL STPTSK(SHOPS(1,SHOP,BASE), PEOPLE(1,3,BASE),
                AGESTK(1,2,BASE), NF, STRIP )
        CALL ACWAIT(SHOPS(1,SHOP,BASE), NO, TASK, NA, PART, NO, NO, NO,
                NO, NOW, NOW, NO)
        CALL BENDAC(NA,TYPE,BASE)
        CALL STTASK(NA, TASK, 1, BASE)
        CALL ENDAC(NA,TYPE,BASE,1)
        CALL REORGN( BASE, KPART )
        CALL ASSET2(BASE)
SUBROUTINE REORGN(BASE,KPART)
        CALL LOSSES(AVAIL,PCT,LOST,NONUNI)
        CALL AISREP(AIS, NO, STRING, 1, BASE)
        CALL ORDER(BASE, N2, AGE, TOT)
        CALL PICK(BASE,BLDG,FRAC(1),PEOLOS,AGELOS,PRTLOS,TEST)
        CALL STPREP(SHOPS(1,SHOP,BASE),PEOPLE(1,3,BASE),AGESTK(1,2,BASE),
                NK, BASE)
        CALL MODHEP(LIMBO(1,4),NNLIMB,NEX,I,ADD)
        CALL EXHEAP(LIMBO(1,4),NNLIMB,0,NEX,FEX,LEX, I )
        CALL REORG2 ( BASE, KPART, FRAC(1) )
SUBROUTINE REORG2 ( BASE, KPART, FRAC )
        CALL LOSSES (STORE, PRTLOS, LST, NONUNI )
        CALL ORDER(BASE,N3,PART,LOST)
        CALL PICK(BASE,BLDG,FRAC(1),PEOLOS,AGELOS,PRTLOS,TEST)
        CALL OUTWAT( SHOPS(16,SHOP,BASE), SHOPS(14,SHOP,BASE),
                SHOPS(15,SHOP,BASE), NT )
        CALL OUTINT(SHOPS(13,SHOP,BASE),SHOPS(11,SHOP,BASE),
                SHOPS(12,SHOP,BASE), NT )
        CALL STPBLD(NF,BASE)
        CALL REDPEO(XPEOP,NUM,BASE,SHOPS(1,1,BASE),PEOPLE(1,1,BASE),
                AGESTK(1,1,BASE),MATERL(1,BASE)  )
        CALL ADDAGE(AGE, NO, BASE)
            IF (CEWORK .EQ. 1)   CALL REBILD(BASE)
        CALL SHCIRF(SHOP,BASE)
        CALL SHPRES(N3,-UNIT,N1,BASE,MMAXB,BASE)
        CALL OUTREP(ATTACK(1,1),LTATT,1,NOATT,FE,LE,PO,TIME)
        CALL INHEAP(CEJOBQ(1,7),LTHCQ,0,NEC,FEC,LEC,ENDTME)
SUBROUTINE PICK(BASE,ARG,FRAC,PEOLOS,AGELOS,PRTLOS,TEST)
SUBROUTINE REBILD(BASE)
        CALL INICON(BLDG,TYPE,BASE,N,START,WHY)
        CALL WAITS(WHY, BASE, TIME)
SUBROUTINE INICON(BLDG, TYPE, BASE, N, START, WHY )
        CALL INHEAP(CEJOBQ(1,7),LTHCQ,0,NEC,FEC,LEC,ENDTME)
        CALL DATE(NOW,DY1,HR1)
```

```
SUBROUTINE BSEREP
    ENTRY ENDCE(LR)
        CALL ADMINI(UNIT, NO, BASE, BASE, NO)
        CALL OUTHEP(CEJOBQ(1,7),LTHCQ,0,NEC,FEC,LEC,PO,TIME)
        CALL DATE(NOW,DY,HR)
        CALL CHECK(BLDG, NO,NO,NO,NO,BASE, NO)
        CALL DOWPF(BASE,N27)
            IF (BLDG .EQ. 30)  CALL DOWBLD(BASE,FLAG)
            IF (EMERG .NE. 0)  CALL GOHOME( NO, BASE )
        CALL INICON(BLDG, TYPE, BASE, N, START, WHY)
        CALL WAITS(WHY,BASE,TIME)
SUBROUTINE ENDAC(NAC,ACTYPE,BASE,ARG1)
    ENTRY BENDAC(NAC,ACTYPE,BASE)
        CALL ENDALT(NAC, ACTYPE, BASE, NO, NO )
        CALL LANDAC(NAC,N1,BASE,BASE)
        CALL EXHEAP(ACN(1,18),MAXACN,1,NEH,NO,NO,NAC)
        CALL REDNOR(PART,NAC,BASE,NO,NO)
        CALL REMTSK(NAC,NF,1,NO,NO )
        CALL OUTINT(SHOPS(5,SHOP,BASE),SHOPS(3,SHOP,BASE),
                SHOPS(4,SHOP,BASE),NP)
        CALL OUTWAT(SHOPS(8,SHOP,BASE), SHOPS(6,SHOP,BASE),
                SHOPS(7,SHOP,BASE), NP)
        CALL STPTSK(SHOPS(1,SHOP,BASE),PEOPLE(1,3,BASE),
                AGESTK(1,2,BASE) , NP, CFLAG)
        CALL KILLAC(NAC,KILL)
SUBROUTINE KILLAC(NAC,KILL)
        CALL REDNOR(PART,NAC,BASE,NO,NO)
        CALL REMTSK(NAC,NF,2,TASK,BASE)
            IF (REFILL(1,CLASS) .GE. 100) CALL ORDER(BASE,CLASS,ACTYPE,N1)
SUBROUTINE SHCIRF(SHOP,BASE)
        CALL SHPRES(NN,-PART,N1,BASE,MMAXB,BASE)
        CALL OUTINT(SHOPS(13,SHOP,BASE),SHOPS(11,SHOP,BASE),
                SHOPS(12,SHOP,BASE), JOBP )
        CALL OUTWAT(SHOPS(16,SHOP,BASE),SHOPS(14,SHOP,BASE),
                SHOPS(15,SHOP,BASE),JOBP )
```

TAB I

```
SUBROUTINE SCSHIP
        CALL TTIME(T,TD,TIME1,NO,NO,NO,SEEDED(2) )
        CALL MODHEP(PERIOD(1,1),20,NE,MC,SUB)
SUBROUTINE SHPRES(CLASS,TYPE,NUM,BASE,NXBASE,OBASE)
        CALL REDPEO(TYPE,NUM,BASE,SHOPS(1,1,BASE),PEOPLE(1,1,BASE),
                AGESTK(1,1,BASE),MATERL(1,BASE) )
        CALL ADDAGE(TYPE, -NUM, BASE)
        CALL ORDER(DESTIN,CLASS,TYPE,NUM)
SUBROUTINE ORDER(BASE,CLASS,KIND,NUM)
        CALL INHEAP(RESUPP(1,3),LLGQ,0,NEG,FEG,LEG,ARRIVE)
        CALL TTIME(TIM,TD,TIME,NO,NO,NO,NO)
SUBROUTINE DOSHIP(TIME,FLAG)
    ENTRY RECSUP
    ENTRY REFIL(NP)
        CALL REDPEO(PEOP, -NUM,BASE,SHOPS(1,1,BASE),PEOPLE(1,1,BASE),
                AGESTK(1,1,BASE),MATERL(1,BASE) )
        CALL CHECK(NO,NO,NO,PEOP,NO,BASE,CFLAG)
        CALL DOWPF(BASE,SHOP)
        CALL ADMINI( -TYPE, NO, OBASE, BASE, NO )
        CALL ADDAGE(TYPE, NUM, BASE)
        CALL CHECK(NO,NO,NO,NO,TYPE,BASE,CFLAG)
        CALL SHPRES(N3,-PART,N1,BASE,NXBASE,OBASE)
        CALL ORDER( DESTIN, N3, UNIT, N1)
        CALL SEND(PART,BASE,RETAIN)
        CALL DOWPF(BASE,SHOP)
        CALL DOWBLD(BASE, FLAG)
        CALL DOWPF(BASE,SHOP)
        CALL PSTFLT(NAC,NO,NO,NO,N1,NO)
        CALL NEWAC(NAC)
        CALL OUTHEP(RESUPP(1,3),LLGQ,0,NEG,FEG,LEG,NP,TIME)
SUBROUTINE STATUS
        CALL INHEAP(REPORT(1,1),NOREPT,1,NORPT,FE,LE,TIME)
        CALL TTIME(XMITM, TD, XMITME, NO, NO, NO, SEEDED(3) )
    ENTRY SNDRPT(LOC)
        CALL OUTHEP(REPORT(1,1),NOREPT,1,NORPT,FE,LE,LOC,TIME)
        CALL INHEAP(REPORT(1,1),NOREPT,1,NORPT,LOC,LE,TIME)
    ENTRY RECRPT(LOC)
        CALL OUTHEP(REPORT(1,1),NOREPT,1,NORPT,FE,LE,LOC,TIME)
        CALL TTIME(XMITM, TD, XMITME, NO, NO, NO, SEEDED(3) )
        CALL INHEAP(REPORT(1,1),NOREPT,1,NORPT,LOC,LE,TIME)
```

TAB J

```
SUBROUTINE SHIFT
        CALL STPREP(SHOPS(1,SHOP,BASE),PEOPLE(1,3,BASE),
                    AGESTK(1,2,BASE),NP,BASE)
             IF (SFLAG .EQ. 1)  CALL STPTSK(SHOPS(1,SHOP,BASE),
                    PEOPLE(1,3,BASE), AGESTK(1,2,BASE), NP, CFLAG )
             IF (SFLAG .EQ. 1)  CALL STPBLD(NP,BASE)
             IF ((CKALT.EQ.1).AND.(ALERTR.EQ.1))  CALL RELALT(BASE,MOD(1))
        CALL CHECK(SHOP,NO,NO,NO,NO,BASE,NO)
        CALL DOWPF(BASE,SHOP)
        CALL DOWBLD(BASE, BFLAG)
SUBROUTINE REDPEO(PEOP,NUM,BASE,SHOPS,PEOPLE,AGESTK,MATERL)
        CALL STPREP(SHOPS(1,SHOP),PEOPLE(1,3), AGESTK(1,2), NP, BASE)
        CALL STPBLD(NP,BASE)
        CALL STPTSK(SHOPS(1,SHOP), PEOPLE(1,3), AGESTK(1,2), NP, CFLAG)
        CALL EXHEAP(CEJOBQ(7,1),LTHCEQ,0,NEC,FEC,LEC,NR)
SUBROUTINE CHECK(ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,FLAG)
        CALL RETASK(SHOPS(1,1,BASE),PEOPLE(1,1,BASE),
                    AGESTK(1,1,BASE),TASKP,START)
        CALL DOWTSK(SHOPS(1,1,BASE),PEOPLE(1,1,BASE),PARTS(1,1,1),
                AGESTK(1,1,BASE),MUNSTK(1,1,BASE),TRAP(1,BASE),TASKP,START)
        CALL REREP(SHOPS(1,1,BASE),PEOPLE(1,3,BASE),AGESTK(1,2,BASE),
                    PARTS(1,1,1),TASKP, BASE)
        CALL DOWREP(SHOPS(1,1,BASE),PEOPLE(1,3,BASE),AGESTK(1,2,BASE),
                    PARTS(1,1,1), TASKP, BASE)
        CALL NEWTSK(SHOPS(1,1,BASE),PEOPLE(1,1,BASE),PARTS(1,1,1),
                    AGESTK(1,1,BASE),MUNSTK(1,1,BASE),TRAP(1,BASE),
                    NA, TASK, RTASK, START, BEGUN  )
        CALL REMTSK(NA,TASKP,2,TSK,BASE)
        CALL DOWPF(BASE,SHOP)
    ENTRY ZCHECK
SUBROUTINE STRTSK
    ENTRY STTASK (NAC, TASK, NFLAG, BASE)
    ENTRY REMTSK(NAC, NP, NFLAG, TASK,BASE)
SUBROUTINE NORRPT
    ENTRY RPTNOR(PART,CRIT,NAC,BASE)
        CALL ORDERP(PART,BASE,FLAG)
        CALL GETPRT(PART,BASE)
        CALL FINDPT(PART,BASE)
    ENTRY REDNOR(PART,NAC,BASE,P1,P2)
SUBROUTINE AVGTME( AVGTSK, AVGREP )
        CALL CKNET( ACTYPE, TASK, PART, PTASK, NR, LEVEL, VERIFY, FLAG )
SUBROUTINE INTRUP
    ENTRY ININT (NE, NF, NL, TIME, NLE)
    ENTRY OUTINT(NE, NF, NL, NO)
SUBROUTINE WAIT
    ENTRY INWAIT(NE,NF,NL,TIME,NLE)
    ENTRY OUTWAT(NE,NF,NL,NO)
SUBROUTINE HEAP
    ENTRY INHEAP(A,NH, NFLAG,NE,FE,LE,TIME)
    ENTRY OUTHEP(A,NH,NFLAG,NE,FE,LE,PO,TIME)
```

```
    ENTRY EXHEAP(A, NH, NFLAG, NE, FE, LE, PR)
    ENTRY MODHEP(A, NH, NE, MC, ADD)
SUBROUTINE ACWAIT(SHOPS,FLAG,NTASK,NAC,PART,AGE,PEO,TME,
                  WHY,WHEN,KNOW,RTASK )
      CALL INWAIT(SHOPS(8),SHOPS(6),SHOPS(7),TIME,NLEW)
SUBROUTINE RESET
BLOCK DATA
SUBROUTINE TTIME(T,TD,TT,HURRYX,REDUCX,SAVEX,SEED )
SUBROUTINE SHPRQT(NAC,NRN,SHOP,ACTYPE,TRANS,TSKP,TASK,ND )
    ENTRY SHPRQ( ND, NAC, ACTYPE, TRANS, TASK )
SUBROUTINE BREAK
SUBROUTINE LOSSES (N, IPK, LOST,NONUNI)
FUNCTION RANDG(N)
    ENTRY IRANDG( FF )
    ENTRY CRANDG( FF )
SUBROUTINE ACCRIT
SUBROUTINE LIST1(L,H,R,NAME1,NAME2,N,FLAG)
    ENTRY LIST1E(L,H,R,N,FLAG)
SUBROUTINE LIST2(L,H,R,NAME1,NAME2,N, MB)
      CALL LIST1E(L,H,R,N(1,1,I),I)
SUBROUTINE LIST3(N,M,T,NAME1,NAME2,A)
SUBROUTINE LIST4(H,L,R,NAME1,NAME2,N,FLAG)
    ENTRY LIST4E(H,L,R,N,FLAG)
SUBROUTINE LIST5(H,L,R,NAME1,NAME2,N, MB)
      CALL LIST4E(H,L,R,N(1,1,I),I)
FUNCTION SHOPST(I, SHOP, BASE)
FUNCTION DAY(TIME)
FUNCTION TOD(TIME)
FUNCTION HRMIN(TIME)
SUBROUTINE DATE(TIME,DY,HR)
FUNCTION THF(TIME)
FUNCTION TU(TIME)
SUBROUTINE HELPER (L, NP)
```

# APPENDIX F

## LABELED COMMON USAGE BY SUBROUTINE

This section lists the name of each labeled Common statement used in TSAR. When available in machine readable form, this list is particularly useful for determining all locations that employ a particular Common statement. A machine readable copy of this list will be provided, along with other items, to any agency that acquires a copy of the TSAR sortie generation source decks.

### TAB A

MAIN ROUTINE
SUBROUTINE TRIALS
- BASIC
- JOBS
- INFO
- REQTS

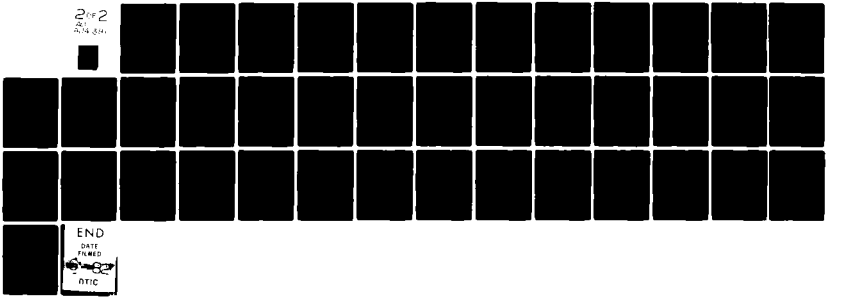
SUBROUTINE MANAGE
- BASIC
- STOCKS
- LOAD
- THEATR
- BOMBSE
- INFO
- REQTS
- AISCOM

SUBROUTINE ADAPT
- BASIC
- STOCKS
- LOAD
- JOBS
- THEATR
- BOMBSE
- INFO
- OUT
- REQTS
- CPARTS
- AISCOM

SUBROUTINE CONTRL
- BASIC
- STOCKS
- LOAD
- JOBS
- THEATR

```
    BOMBSE
    REQTS
    INFO
    OUT
    LOCAL4
SUBROUTINE FERRY
    BASIC
    JOBS
    LOAD
    STOCKS
    REQTS
    BOMBSE
SUBROUTINE GOHOME
    BASIC
    LOAD
```

AD-A114 391 RAND CORP SANTA MONICA CA F/G 15/3
TSAR USER'S MANUAL. VOLUME III. VARIABLE AND ARRAY DEFINITIONS,--ETC(U)
FEB 82 D E EMERSON F49620-82-C-0018
UNCLASSIFIED RAND/N-1822-AF NL



TAB B

```
SUBROUTINE INIT
   BASIC
   STOCKS
   LOAD
   JOBS
   THEATR
   BOMBSE
   INFO
   OUT
   REQTS
   CPARTS
   AISCOM
   PURGE1
   PURGE2
   PURGE3
   TESTS
SUBROUTINE INPUT
   BASIC
   STOCKS
   LOAD
   JOBS
   THEATR
   BOMBSE
   INFO
   OUT
   REQTS
   CPARTS
   PURGE1
   LOCAL1
   TESTS
   TIMHOR
SUBROUTINE INPUTA
   BASIC
   STOCKS
   LOAD
   JOBS
   THEATR
   BOMBSE
   INFO
   OUT
   CPARTS
   AISCOM
   PURGE1
SUBROUTINE INPUTB
   BASIC
   STOCKS
   LOAD
   JOBS
   THEATR
   BOMBSE
```

```
    BOMBSE
    REQTS
    INFO
    OUT
    LOCAL4
SUBROUTINE FERRY
    BASIC
    JOBS
    LOAD
    STOCKS
    REQTS
    BOMBSE
SUBROUTINE GOHOME
    BASIC
    LOAD
```

```
    INFO
    OUT
    CPARTS
SUBROUTINE INPUTC
SUBROUTINE WRAPUP
    BASIC
    STOCKS
    JOBS
    LOAD
    THEATR
    BOMBSE
    REQTS
    INFO
    CPARTS
    AISCOM
    PURGE1
SUBROUTINE ICHECK
    BASIC
    STOCKS
    JOBS
    LOAD
    CPARTS
    OUT
    PURGE1
    LOCAL1
SUBROUTINE HELPCK
    BASIC
    STOCKS
    LOCAL1
SUBROUTINE INLIST
    BASIC
    STOCKS
    LOAD
    JOBS
    THEATR
    BOMBSE
    INFO
    OUT
SUBROUTINE HEADER
    BASIC
    JOBS
    THEATR
    BOMBSE
    INFO
    CPARTS
    REQTS
    TESTS
SUBROUTINE INITIZ
    BASIC
    STOCKS
    JOBS
    LOAD
```

```
   THEATR
   BOMBSE
   INFO
   REQTS
SUBROUTINE ZSHOPS
   BASIC
   STOCKS
   JOBS
   REQTS
   PURGE5
SUBROUTINE TESTER
   BASIC
   BOMBSE
   AISCOM
SUBROUTINE MODIFY
   BASIC
   JOBS
SUBROUTINE CKNET
   JOBS
SUBROUTINE CKRQT
   JOBS
   REQTS
   PURGE1
   PURGE2
SUBROUTINE CKSPLT
   BASIC
   JOBS
   REQTS
   PURGE1
SUBROUTINE ZNOR
   BASIC
   STOCKS
   JOBS
   ZZNORS
```

TAB C

```
SUBROUTINE COMPRT
   BASIC
   STOCKS
   JOBS
   THEATR
   CPARTS
   PURGE1
   PURGE3
SUBROUTINE IPARTS
   BASIC
   STOCKS
   JOBS
   THEATR
   REQTS
   CPARTS
   PURGE1
   PURGE3
SUBROUTINE IPART2
   BASIC
   STOCKS
   THEATR
   CPARTS
   PURGE1
   PURGE3
   ZZNORS
SUBROUTINE CKNRTS
   BASIC
   STOCKS
   JOBS
   THEATR
   CPARTS
   PURGE3
SUBROUTINE KREQTS
   BASIC
   JOBS
   REQTS
SUBROUTINE REQTS1
   BASIC
   STOCKS
   JOBS
   REQTS
SUBROUTINE OUTPUT
   BASIC
   STOCKS
   JOBS
   THEATR
   BOMBSE
   AISCOM
   OUT
   LOCAL2
```

```
SUBROUTINE ASSETS
   BASIC
   STOCKS
   CPARTS
SUBROUTINE ASSET2
   BASIC
   STOCKS
SUBROUTINE TIMES
   BASIC
   OUT
SUBROUTINE DELAYS
   BASIC
   STOCKS
   JOBS
   BOMBSE
   CPARTS
   OUT
   AISCOM
   LOCAL2
SUBROUTINE PSHORT
SUBROUTINE JOBLST
SUBROUTINE DEFERS
   BASIC
   JOBS
```

TAB D

```
SUBROUTINE READFT
   BASIC
   LOAD
   THEATR
   BOMBSE
   REQTS
   LOCAL3
SUBROUTINE FRAG
   BASIC
   BOMBSE
   REQTS
SUBROUTINE PLAN
   BASIC
   STOCKS
   LOAD
   JOBS
   BOMBSE
SUBROUTINE PLAN1
   BASIC
SUBROUTINE BASCAP
   BASIC
   LOAD
   STOCKS
   JOBS
   INFO
   BOMBSE
   REQTS
   LOCAL3
SUBROUTINE REASSG
   BASIC
   STOCKS
   LOAD
   JOBS
   REQTS
   CPARTS
   LOCAL3
SUBROUTINE FLYERS
   BASIC
SUBROUTINE DISABL
   BASIC
SUBROUTINE FLIGHT
   BASIC
   LOAD
   OUT
SUBROUTINE LAUNCH
   BASIC
   STOCKS
   OUT
   REQTS
SUBROUTINE SORT
   BASIC
```

TAB E

```
SUBROUTINE PSTFLT
    BASIC
    STOCKS
    LOAD
    JOBS
    BOMBSE
    OUT
SUBROUTINE RUNAC
    BASIC
    STOCKS
    LOAD
    JOBS
    BOMBSE
    OUT
SUBROUTINE INITSK
    BASIC
    JOBS
    THEATR
    BOMBSE
    OUT
SUBROUTINE DOTASK
    BASIC
    JOBS
SUBROUTINE ENDTSK
    BASIC
    JOBS
SUBROUTINE CHKWX
    BASIC
SUBROUTINE INIDEF
    BASIC
    STOCKS
    JOBS
    CPARTS
SUBROUTINE CANNIB
    BASIC
    STOCKS
    JOBS
SUBROUTINE INCOMP
    BASIC
    JOBS
SUBROUTINE CKCRIT
    BASIC
    STOCKS
    JOBS
    BOMBSE
SUBROUTINE CKROOT
    BASIC
SUBROUTINE SCHJOB
    BASIC
    STOCKS
```

```
    JOBS
SUBROUTINE SPLIT
    BASIC
SUBROUTINE GETPEO
    BASIC
    JOBS
```

TAB F

```
SUBROUTINE PREFLT
    BASIC
    STOCKS
    LOAD
    JOBS
    LDAMMO
    RECNF
SUBROUTINE ASSIGN
    BASIC
    STOCKS
SUBROUTINE RECNFG
    BASIC
    LOAD
    OUT
    RECNF
SUBROUTINE UPLOAD
    BASIC
    OUT
    LDAMMO
SUBROUTINE REFUEL
    BASIC
    JOBS
    OUT
SUBROUTINE DOWPRE
    BASIC
    STOCKS
    LOAD
    JOBS
    OUT
    LDAMMO
    RECNF
SUBROUTINE MUNEED
    BASIC
    STOCKS
    LOAD
    JOBS
    OUT
SUBROUTINE CKBILD
    BASIC
    JOBS
    STOCKS
SUBROUTINE DOBILD
    BASIC
    STOCKS
    JOBS
    OUT
SUBROUTINE CKPEOP
    BASIC
SUBROUTINE CKAGE
    BASIC
```

```
    JOBS
SUBROUTINE ADDAGE
    BASIC
    STOCKS
    JOBS
    AISCOM
SUBROUTINE CKALRT
    BASIC
    STOCKS
    JOBS
SUBROUTINE RELALT
    BASIC
    STOCKS
    LOAD
```

TAB G

```
SUBROUTINE ADMIN
    BASIC
    STOCKS
    JOBS
    INFO
    THEATR
    BOMBSE
    AISCOM
SUBROUTINE RUNSHP
    BASIC
    STOCKS
    JOBS
    THEATR
    OUT
SUBROUTINE INIREP
    BASIC
    JOBS
    THEATR
    INFO
    OUT
    AISCOM
SUBROUTINE DOREP
    BASIC
    JOBS
    AISCOM
SUBROUTINE ENDREP
    BASIC
    JOBS
    AISCOM
SUBROUTINE SALVAG
    BASIC
    JOBS
    THEATR
SUBROUTINE CKCIRF
    BASIC
    STOCKS
    JOBS
SUBROUTINE REPRTY
    BASIC
    STOCKS
    JOBS
    OUT
```

SUBROUTINE CKAIS
BASIC
OUT
JOBS
AISCOM
SUBROUTINE NRTSIT
BASIC
STOCKS
THEATR

TAB H

```
SUBROUTINE BOMB
   BASIC
   STOCKS
   JOBS
   BOMBSE
   AISCOM
   PURGE4
SUBROUTINE REORGN
   BASIC
   STOCKS
   JOBS
   THEATR
   BOMBSE
   OUT
   AISCOM
   PURGE4
SUBROUTINE REORG2
   BASIC
   STOCKS
   JOBS
   THEATR
   BOMBSE
   OUT
   AISCOM
   PURGE4
SUBROUTINE PICK
   BOMBSE
SUBROUTINE REBILD
   BASIC
   STOCKS
   BOMBSE
SUBROUTINE INICON
   BASIC
   STOCKS
   BOMBSE
SUBROUTINE BSEREP
   BASIC
   STOCKS
   JOBS
   BOMBSE
SUBROUTINE ENDAC
   BASIC
   STOCKS
   LOAD
   JOBS
SUBROUTINE KILLAC
   BASIC
   JOBS
SUBROUTINE SHCIRF
   BASIC
```

STOCKS
JOBS
THEATR
BOMBSE

TAB I

SUBROUTINE SCSHIP
- BASIC
- THEATR

SUBROUTINE SHPRES
- BASIC
- STOCKS
- THEATR

SUBROUTINE ORDER
- BASIC
- STOCKS

SUBROUTINE DOSHIP
- BASIC
- STOCKS
- JOBS
- THEATR
- BOMBSE
- CPARTS

SUBROUTINE STATUS
- BASIC
- STOCKS
- INFO

TAB J

```
SUBROUTINE SHIFT
   BASIC
   STOCKS
   BOMBSE
SUBROUTINE REDPEO
   BASIC
   BOMBSE
SUBROUTINE CHECK
   BASIC
   STOCKS
   JOBS
   LOCAL3
SUBROUTINE STRTSK
   BASIC
   STOCKS
   JOBS
SUBROUTINE NORRPT
   BASIC
   STOCKS
SUBROUTINE AVGTME
   BASIC
   STOCKS
   REQTS
   JOBS
   CPARTS
SUBROUTINE INTRUP
   BASIC
SUBROUTINE WAIT
   BASIC
SUBROUTINE HEAP
SUBROUTINE ACWAIT
   BASIC
SUBROUTINE RESET
   BASIC
   THEATR
   BOMBSE
   INFO
   BASIC
SUBROUTINE TTIME
SUBROUTINE SHPRQT
   BASIC
   JOBS
SUBROUTINE BREAK
   BASIC
   JOBS
   OUT
SUBROUTINE LOSSES
SUBROUTINE ACCRIT
   BASIC
   STOCKS
   JOBS
```

```
SUBROUTINE LIST1
SUBROUTINE LIST2
SUBROUTINE LIST3
SUBROUTINE LIST4
SUBROUTINE LIST5
    STOCKS
SUBROUTINE DATE
    TIMHOR
    TIMHOR
SUBROUTINE HELPER
    BASIC
    STOCKS
    BOMBSE
    OUT
```

# APPENDIX G

## JOB CONTROL LANGUAGE CONTROL CARDS

### Control Cards to Execute TSAR

The Job Control Language cards required to execute the TSAR simulation when TSARINA is not used during the same computer run are illustrated below. Air base attack data for the first trial are entered in their appropriate location; if different damage data are to be used for subsequent trials the data for the first trial should be concluded with a special Card Type #40 with '999' entered in columns 3-5, as explained in the comments that will be found with the source code for Subroutine INPUTC. The damage data for the subsequent trials are then entered following the last of the sortie demand data (Card Types #50) and demand terminator cards (Card Type #99) that will be read in during the simulation. The damage data for each trial other than the first must be terminated by entering a card with ' 0' in columns 1-2.

```
//X0000SET  JOB  (0000,250,7,30),'JCL TO EXECUTE TSAR 1',CLASS=N
//JOBLIB  DD  DSN=X.X0000.A0000.TSAR.I.EXECUTE,DISP=SHR
//GO PROC
//GO   EXEC   PGM=TSAR#I#1
//GO.FT05F001  DD  DDNAME=SYSIN
//GO.FT06F001  DD  SYSOUT=A
//GO.FT07F001  DD  SYSOUT=B
//GO.FT10F001  DD  UNIT=TEMP,SPACE=(TRK,(40,2)),
//    DCB=(RECFM=VS,BLKSIZE=10000),DISP=(NEW,PASS)
//GO.FT11F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//GO.FT12F001  DD  UNIT=TEMP,SPACE=(TRK,(10,2)),
//    DCB=(RECFM=VS,BLKSIZE=5000),DISP=(NEW,PASS)
//GO.FT15F001  DD  UNIT=TEMP,SPACE=(TRK,(40,4)),
//    DCB=(RECFM=VS,BLKSIZE=10000),DISP=(NEW,PASS)
//*       1         2         3         4         5         6         7         8
//*  5678901234567890123456789012345678901234567890123456789012345678901234567890
//*  ********************************************************************************
// PEND
//STEP1 EXEC GO,REGION.GO=1340K
//GO.SYSIN   DD   *

          The TSAR input deck is to be entered here.
/*
```

Control Cards for Executing TSARINA and TSAR

The Job Control Language cards shown below are illustrative of the JCL requirements for executing TSARINA and TSAR together.

```
//X0000#01  JOB    (0000,200,7,20),'TSAR.TSARINA.JCL',CLASS=N
//STEP1     EXEC   PGM=OBJ#1#81,REGION=280K
//STEPLIB   DD     DSN=X.X0000.A0000.TSARIN.OBJECT,DISP=SHR
//FT05F001  DD   DDNAME=SYSIN
//FT06F001  DD   SYSOUT=A
//FT07F001  DD   SYSOUT=B
```

The following ten cards create ten separate temporary data sets in which the results for each of ten trial calculations of the airbase attacks are stored. If more trials are used, more data sets would be required.

```
//FT21F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//FT22F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//FT23F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//FT24F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//FT25F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//FT26F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//FT27F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//FT28F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//FT29F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//FT30F001  DD  UNIT=TEMP,SPACE=(TRK,(1,1)),DISP=(NEW,PASS)
//GO.SYSIN  DD  *
```

The input data for the TSARINA calculations are entered here. The CONTrol card is entered first, followed by the DATA card and other basic control cards, the target cards, attack cards and weapon effectiveness cards; the last entry card should be an END card.

```
/*
```

The JCL cards on the next page complete the requirements for a complete TSARINA-TSAR simulation.

```
//STEP2      EXEC     PGM=TSAR#I#1,REGION=1320K
//STEPLIB     DD      DSN=X.X0000.A0000.TSAR.I.EXECUTE,DISP=SHR
//FT05F001    DD      DDNAME=SYSIN
//FT06F001    DD      SYSOUT=A
//FT07F001    DD      SYSOUT=B
//FT10F001    DD      UNIT=TEMP,SPACE=(TRK,(40,2)),
//    DCB=(RECFM=VS,BLKSIZE=10000),DISP=(NEW,PASS)
//FT11F001    DD      UNIT=TEMP,SPACE=(TRK,(1,1)),DIST=(NEW,PASS)
//FT12F001    DD      UNIT=TEMP,SPACE=(TRK,(10,2)),
//    DCB=(RECFM=VS,BLKSIZE=10000),DISP=(NEW,PASS)
//FT15F001    DD      UNIT=TEMP,SPACE=(TRK,(40,4)),
//    DCB=(RECFM=VS,BLKSIZE=10000),DISP=(NEW,PASS)
//FT21F001    DD      DSN=*.STEP1.FT21F001,DISP=(OLD,DELETE)
//FT22F001    DD      DSN=*.STEP1.FT22F001,DISP=(OLD,DELETE)
//FT23F001    DD      DSN=*.STEP1.FT23F001,DISP=(OLD,DELETE)
//FT24F001    DD      DSN=*.STEP1.FT24F001,DISP=(OLD,DELETE)
//FT25F001    DD      DSN=*.STEP1.FT25F001,DISP=(OLD,DELETE)
//FT26F001    DD      DSN=*.STEP1.FT26F001,DISP=(OLD,DELETE)
//FT27F001    DD      DSN=*.STEP1.FT27F001,DISP=(OLD,DELETE)
//FT28F001    DD      DSN=*.STEP1.FT28F001,DISP=(OLD,DELETE)
//FT29F001    DD      DSN=*.STEP1.FT29F001,DISP=(OLD,DELETE)
//FT30F001    DD      DSN=*.STEP1.FT30F001,DISP=(OLD,DELETE)
//GO.SYSIN    DD      *
```

The TSAR input data deck is entered here. The airbase attack damage data will be read in from the temporary data sets at the appropriate time during the simulation when an '888' is entered in columns 3-5 on a Card Type #40.

If '777' is entered on that card, rather than '888', the damage data stored for the first trial by TSARINA will be used for all trials in the TSAR simulation. Otherwise, all TSAR input procedures are unaffected when TSARINA is used with TSAR.

```
/*
```

Control Cards for Compiling TSAR

The Job Control Language cards shown below can be used to compile the TSAR source code, or selected portions of that code. As noted, the Common statements are stored as members of the partitioned data set TSAR.I.COMMON and are entered into the appropriate subroutine source codes using JCL statements that will be found near the beginning of the various subroutines. Two options for an OVERLAY structure are illustrated.

```
//X0000#01   JOB   (0000,40,7,40),'COMPILE.TSAR.I',CLASS=F
//STEP1      EXEC  PGM=IEFBR14
//NEWMOD  DD  DSN=&&TEMP,DISP=(NEW,PASS),UNIT=TEMP,
//            DCB=(RECFM=U,LRECL=3156,BLKSIZE=3156),
//            SPACE=(TRK,(60,10,1))
//BASIC       DD    DSN=X.X0000.A0000.TSAR.I.COMMON(BASIC),DISP=SHR
//STOCKS      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(STOCKS),DISP=SHR
//LOAD        DD    DSN=X.X0000.A0000.TSAR.I.COMMON(LOAD),DISP=SHR
//JOBS        DD    DSN=X.X0000.A0000.TSAR.I.COMMON(JOBS),DISP=SHR
//THEATR      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(THEATR),DISP=SHR
//BOMBSE      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(BOMBSE),DISP=SHR
//REQTS       DD    DSN=X.X0000.A0000.TSAR.I.COMMON(REQTS),DISP=SHR
//CPARTS      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(CPARTS),DISP=SHR
//INFO        DD    DSN=X.X0000.A0000.TSAR.I.COMMON(INFO),DISP=SHR
//OUT         DD    DSN=X.X0000.A0000.TSAR.I.COMMON(OUT),DISP=SHR
//AISCOM      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(AISCOM),DISP=SHR
//PURGE1      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(PURGE1),DISP=SHR
//PURGE2      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(PURGE2),DISP=SHR
//PURGE3      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(PURGE3),DISP=SHR
//PURGE4      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(PURGE4),DISP=SHR
//PURGE5      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(PURGE5),DISP=SHR
//LOCAL1      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(LOCAL1),DISP=SHR
//LOCAL2      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(LOCAL2),DISP=SHR
//LOCAL3      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(LOCAL3),DISP=SHR
//LOCAL4      DD    DSN=X.X0000.A0000.TSAR.I.COMMON(LOCAL4),DISP=SHR
//*
//STEP2      EXEC   FORTHCL,REGC=400K,PARMC='XREF,OPT(2),NOTERM',
//       LIBL='SYS1.FORTERR',PARM.LKED='OVLY,MAP,XCAL'
//FORT.SYSIN  DD    *,DCB=BLKSIZE=800
```

The TSAR source decks are placed here. Two versions of the concluding JCL cards appear on subsequent pages, illustrating the two options for an OVERLAY structure.

OVERLAY Option Number 1

The following OVERLAY structure has been designed to minimize the core requirement without excessive or overly inefficient movement of the segments. The second overlay structure is even more efficient but takes somewhat more core.

```
//LKED.SYSLMOD    DD    DSN=&&TEMP(TSAR#1#1),DISP=(OLD,PASS)
//LKED.OLDLIB     DD    DSN=X.X0000.A0000.TSAR.1.EXECUTE,DISP=SHR
//LKED.SYSIN  DD  *
   ENTRY MAIN
   INCLUDE OLDLIB(TSAR#1#1)
            INSERT MAIN,TRIALS,MODIFY
            INSERT SHPRQT,CKNET,TTIME
   OVERLAY ZERO
            INSERT INIT,INPUT,CKRQT,PURGE1,PURGE2,PURGE3,LOCAL1
      OVERLAY ONE
            INSERT INPUTA,INPUTB,INPUTC,TESTER,ENTRYP
      OVERLAY ONE
            INSERT WRAPUP,PSHORT
         OVERLAY TWO
            INSERT ICHECK,HELPCK,CKSPLT
         OVERLAY TWO
            INSERT COMPRT,IPARTS,IPART2,CKNRTS
         OVERLAY TWO
            INSERT INITIZ,INLIST,HEADER,AVGTME,RREQTS,REQTS1
            INSERT LIST1,LIST2,LIST3,LIST4,LIST5
   OVERLAY ZERO
            INSERT MANAGE,ADMIN,PLAN,PLAN1,SORT
            INSERT SHIFT,REDPEO,CKPEOP,CKAIS,DOBILD
            INSERT CONTRL,TIMES,ASSET2,HELPER,LOCAL3,LOCAL4
      OVERLAY THREE
            INSERT READFT,FRAG,SCSHIP,ADAPT,MUNEED,CKBILD
            INSERT OUTPUT,ASSETS,ACCRIT,BREAK,DELAYS,JOBLST,LOCAL2
      OVERLAY THREE
            INSERT FERRY,GOHOME,PSTFLT,RUNAC,INITSK,DOTASK,ENDTSK
            INSERT CHKWX,INCOMP,INIDEF,CANNIB,SCHJOB,SPLIT,CKROOT
            INSERT RUNSHP,INIREP,DOREP,ENDREP,SALVAG,REPRTY,NRTSIT
            INSERT STATUS,CHECK,STRTSK,NORRPT,INTRUP,WAIT,ACWAIT
            INSERT DISABL,GETPEO,CKCRIT,CKAGE,ADDAGE,CKALRT,RELALT
            INSERT ENDAC,KILLAC,DOSHIP,SHPRES,ORDER,SHCIRF,CKCIRF
            INSERT BSEREP,INICON,FTIME
         OVERLAY FOUR
            INSERT ZSHOPS,ZNOR,PURGE5
         OVERLAY FOUR
            INSERT FLYERS,FLIGHT,LAUNCH
            INSERT PREFLT,ASSIGN,RECNFG,UPLOAD,REFUEL,DOWPRE
            INSERT BASCAP,REASSG,RESET
         OVERLAY FOUR
            INSERT BOMB,REORGN,REORG2,PICK,REBILD,LOSSES,PURGE4
```

```
OVERLAY FOUR
   INSERT BOMB,REORGN,REORG2,PICK,REBILD,LOSSES,PURGE4
```

The JCL that concludes the TSAR compile and link-edit process is shown below. These cards carry out the link-editing on a temporary disk, scratch the earlier load module, and restore the new load module on the same disk using the old name; this sequence is used to avoid loss of the load module when space is insufficient on the normal storage disk to temporarily store both the old and new versions.

```
//SCRATCH   EXEC   PGM=IEHPROGM,COND=(4,LT,STEP2.FORT)
//SYSPRINT   DD    SYSOUT=A
//DD1        DD    DSN=X.X0000.A0000.TSAR.I.EXECUTE,DISP=OLD
//SYSIN      DD    *
    SCRATCH  DSNAME=X.X0000.A0000.TSAR.I.EXECUTE,VOL=USER=USER81,      X
              MEMBER=TSAR#I#1
//COPY      EXEC   PGM=IEBCOPY,REGION=280K,COND=(4,LT,STEP2.FORT)
//SYSPRINT   DD    SYSOUT=A
//IN         DD    DSN=&&TEMP,DISP=(OLD,PASS)
//PRESS      DD    DSN=X.X0000.A0000.TSAR.I.EXECUTE,DISP=SHR
//OUT        DD    DSN=X.X0000.A0000.TSAR.I.EXECUTE,DISP=OLD
//SYSUT3     DD    UNIT=TEMP,SPACE=(TRK,(10))
//SYSUT4     DD    UNIT=TEMP,SPACE=(TRK,(10))
//SYSIN      DD    *
   COPY INDD=PRESS,OUTDD=OUT
   COPY INDD=IN,OUTDD=OUT
```

OVERLAY Option Number 2

The second OVERLAY option shown below makes minimal use of overlay but nevertheless achieves a significant reduction in the core requirement by getting rid of the unneeded input and initialization subroutines and data storage when the simulation begins.

```
//LKED.SYSLMOD    DD    DSN=&&TEMP(TSAR#1#1),DISP=(OLD,PASS)
//LKED.OLDLIB     DD    DSN=X.X0000.A0000.TSAR.1.EXECUTE,DISP=SHR
//LKED.SYSIN  DD  *
   ENTRY MAIN
   INCLUDE OLDLIB(TSAR#1#1)
             INSERT MAIN,TRIALS,MODIFY
             INSERT SHPRQT,CKNET,TTIME
   OVERLAY ZERO
             INSERT INIT,INPUT,CKRQT,PURGE1,PURGE2,PURGE3,LOCAL1
             INSERT INPUTA,INPUTB,INPUTC,TESTER,ENTRYP
             INSERT WRAPUP,PSHORT
             INSERT ICHECK,HELPCK,CKSPLT
             INSERT COMPRT,IPARTS,IPART2,CKNRTS
             INSERT INITIZ,INLIST,HEADER,AVGTME,RREQTS,REQTS1
             INSERT LIST1,LIST2,LIST3,LIST4,LIST5
   OVERLAY ZERO
             INSERT MANAGE,ADMIN,PLAN,PLAN1,SORT
             INSERT SHIFT,REDPEO,CKPEOP,CKAIS,DOBILD
             INSERT CONTRL,TIMES,ASSET2,HELPER,LOCAL3,LOCAL4
             INSERT READFT,FRAG,SCSHIP,ADAPT,MUNEED,CKBILD
             INSERT OUTPUT,ASSETS,ACCRIT,BREAK,DELAYS,JOBLST,LOCAL2
             INSERT FERRY,GOHOME,PSTFLT,RUNAC,INITSK,DOTASK,ENDTSK
             INSERT CHKWX,INCOMP,INIDEF,CANNIB,SCHJOB,SPLIT,CKROOT
             INSERT RUNSHP,INIREP,DOREP,ENDREP,SALVAG,REPRTY,NRTSIT
             INSERT STATUS,CHECK,STRTSK,NORRPT,INTRUP,WAIT,ACWAIT
             INSERT DISABL,GETPEO,CKCRIT,CKAGE,ADDAGE,CKALRT,RELALT
             INSERT ENDAC,KILLAC,DOSHIP,SHPRES,ORDER,SHCIRF,CKCIRF
             INSERT BSEREP,INICON,FTIME
             INSERT ZSHOPS,ZNOR,PURGE5
             INSERT FLYERS,FLIGHT,LAUNCH
             INSERT PREFLT,ASSIGN,RECNFG,UPLOAD,REFUEL,DOWPRE
             INSERT BASCAP,REASSG,RESET
             INSERT BOMB,REORGN,REORG2,PICK,REBILD,LOSSES,PURGE4
//SCRATCH    EXEC    PGM=IEHPROGM,COND=(4,LT,STEP2.FORT)
//SYSPRINT    DD     SYSOUT=A
//DD1         DD     DSN=X.X0000.A0000.TSAR.1.EXECUTE,DISP=OLD
//SYSIN       DD     *
    SCRATCH  DSNAME=X.X0000.A0000.TSAR.1.EXECUTE,VOL=USER=USER81,      X
              MEMBER=TSAR#1#1
//COPY       EXEC    PGM=IEBCOPY,REGION=280K,COND=(4,LT,STEP2.FORT)
//SYSPRINT    DD     SYSOUT=A
```

```
//IN          DD     DSN=&&TEMP,DISP=(OLD,PASS)
//PRESS       DD     DSN=X.X0000.A0000.TSAR.I.EXECUTE,DISP=SHR
//OUT         DD     DSN=X.X0000.A0000.TSAR.I.EXECUTE,DISP=OLD
//SYSUT3      DD     UNIT=TEMP,SPACE=(TRK,(10))
//SYSUT4      DD     UNIT=TEMP,SPACE=(TRK,(10))
//SYSIN       DD     *
   COPY  INDD=PRESS,OUTDD=OUT
   COPY  INDD=IN,OUTDD=OUT
```

# APPENDIX H

## SUBROUTINE FOR USE IN CREATING COMMON STATEMENTS

The data set listed in this section has been created to assist TSAR users in creating a new set of labeled Common statements when the TSAR simulation program is to be redimensioned. Each Common statement is reproduced here in the exact form in which it should be stored for retrieval for program compilations, except that each array dimension the user may change is replaced by the variable name for that dimension. The user may create the required Common statements by using his system editor to change all appearances of each variable name to the desired value for that variable.

These conveniences are provided both for systems with a single central core and for CDC systems such as the CYBAR 176 that have both a small, high speed core and a larger, lower speed core. The second half of the data set is for Common for CDC machines. The names chosen for the labeled Common to be placed in the larger, slower core were created by adding an "L" as a prefix to the name of the labeled Common from which they were derived. As will be noted, in some cases the entire labeled Common is to be stored in the larger core; this is done with the REQTS, CPARTS, AISCOM, PURGE1, PURGE2, PURGE3, PURGE4, and PURGE5 Commons.

This data set has been designed to be used with an editor that permits changes to be made on lines that have a particular character in a specified column. For that reason the FORTRAN comment marks--i.e., "C" in column 1--have been retained to make clear how that editing process works.

The exemplary commands described below for carrying out the various steps in this process are appropriate for use on the WYLBUR editing system as used on The Rand Corporation's IBM 370/3033; the requirements for achieving the equivalent effects on other operating systems should be determined by discussions with a system specialist for that system.

The labeled Commons are created with statements such as:

```
CH 'name' TO 'value' in ' '1/1
```

that is, make the change in all lines that have a blank in col 1. When all the variable names have been replaced with the desired values, all other lines are modified with the statement:

```
CH 'N'1/1 TO ' '
```

and the 'N' in col 1 is replaced with a blank.

After specific values have been assigned to all array dimensions, it will be necessary to fix an error that will have been introduced into the name "SCLRQT" in the BASIC3 Common statement.

Then each labeled Common should be stored in the partitioned data set created for that purpose with a command of the type:

```
SAV   module.name(common name)  CARD(10)
```

The necessary PDS can be created with the WYLBUR command:

TIPS ALLOCATE module.name USERxx RECFM=FB LRECL=80 BLK=800 TRK 2 1 25

The labeled Common statements should be created by dimensioning the arrays in the order shown on the next page. The values listed here are those used for a particular TSAR load module created in early 1982.

```
C
C
C
C
N       MAXACN =  400
N       MAXB   =    8
N       MAXT   =    3
N       MAXM   =    4
C
C       NEXT INSERT THE DIMENSIONS FOR THE
C       DYNAMICALLY GENERATED QUEUES AND HEAPS.
C
C
N       LTQ    = 1000
N       LRQ    =  250
N       LIQ    =  400
N       LWQ    = 2000
N       LBQ    =  400
N       LLQ    =  600
N       LTHCEQ =  400
N       LNT    = 2500
N       LDT    = 4000
N       LNOR   = 2000
N       LGQ    = 2000
N       NJOINT =  200
N       NOCHG  =  100
N       NOPKG  = 2500
N       NOSHIP =  400
N       LFQ    =  400
N       MAXPER =  160
C
C       NEXT, ADD THE DIMENSIONS OF THE MISCELLANEOUS DATA ARRAYS
C
N       NOREPT =   50
N       NOCONS =  200
N       NCARGO = 2500
N       NOSHP  =  100
N       LTHATT =   40
N       NOITEM = 2000
```

```
N         LTHLST =   90
N         NOUSER =  120
N         NLIMBO = 1500
N         MAXREC =   50
C      NSCROL MAY NOT BE GREATER THAN 24
N         NSCROL =   24
C      WXDAYS MAY NOT BE GREATER THAN 65
N         WXDAYS =   65
C
C         DIMENSION THE STORAGE ARRAYS FOR THE VARIOUS TASK REQUIREMENTS
C
N         NOTSKA =  200
N         NOTSK  = 1500
N         NOREPA =  200
N         NOREP  = 1000
N         NOAGER =  100
N         NOSCL  =   30
N         NOCONF =   30
N         NOBILD =   30
N         NOCE   =   20
N         NOTASK =   32
C
C         THE RESOURCE STORAGE ARRAYS ARE DIMENSIONED LAST.
C
N         NOCREW =  800
N         NOPEOP =  200
N         NOAGE  =  100
N         NOSTAT =   10
N         NOTRAY =  100
N         NOPART = 1000
N         NOPRT  = 1000
N         EXTPRT =  300
N         NOMUN  =   25
N         NOTRAP =   25
N         NOMATL =   25
N         NOFAC  =  100
C
C     ***************************************************************************
C     ***************************************************************************
C
N      REAL     *4        SEEDS(10)
N      INTEGER *4         MAXACN,MAXB,MAXT,MAXM,NOTSK,NOTSKA,NOREP,NOREPA,
N     X      NOAGER,NOBILD,NOPEOP,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOSCL,
N     X      NOCONF,NOPRT,LTHLST,NXSEED,
      X      LATERL( MAXB ), PRTCRT( NOPRT,2), RINDEX(4,9)
C
C    ** COMMONS BASIC1, BASIC2, BASIC3 AND BASIC4 ARE INCLUDED BY "BASIC"
C
C
N      COMMON  /BASIC1/ MAXACN,MAXB,MAXT,MAXM,NOTSK,NOTSKA,NOREP,NOREPA,
N     X  NOAGER,NOBILD,NOPEOP,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOSCL,
N     X  NOCONF,NOPRT,LTHLST, NXSEED, LATERL, PRTCRT, RINDEX
```

```
C
C   ************           KEY CONTROL VARIABLES                ************
C
N       COMMON / BASIC2 /   NOW, HOUR, NBASE, NTYPE, NPART, CREWS,
N      X  SIMLTH, EXTEND, NTRIAL, ITRIAL, PRINT, STATFQ, CUMSTA, OVERFL,
N      X  TSAR, FIXAGE, BUILD, AIDA, DAMODE, NOSAVE, FSALVG,
N      X  TEST, XTEST, SEED, VERIFY, IDAY, NOUSER, NONUNI, DOWNTM, CANSRU,
N      X  SCROLL, NSCROL, SCROL1, SCROL2, TOOFEW, K1LOW, K2LOW,
N      X  SLEEP, REST, NOCREW, NPILOT, MAXFLT, MAXPER, MULTI, CANFLT,
N      X  NOPOMO,ORDIT,ORDWT,CMODE,NOFUEL,CCIRF,CTHEA,SHOPRY,CIRFLG,
N      X  UNCER, CONSIG, TODOCK, ADAPTR, SHPREP, SEEKSH, NRTPOL, PROTME,
N      X  OPSBSE, EMERG, POSTPN, CONCUR, LTHDEF, ALERTR, DELYPF, OVERTM,
N      X  JOBCON, FILLAC, FLEVEL, MNTLMT, MNTF, MNTR, QUIK, RPARTS,MAXMNT,
N      X  NEWPRT, ZNORS, LOSTAC, MAXTME,
N      X  CANMOD, CANMUL, CANCAN, DOCANN, MXHOLE, CDELAY,
N      X  MAXREC,WXDAYS,TPLAN,NEXTIN,NEXTSC,ENDAY,LOADTM,LSTTOD,DMDLMT,
N      X  SPARE1,SPARE2,SPARE3,SPARE4,SPARE5,SPARE6,SPARE7,SPARE8,SPARE9,
       X  HURRY(MAXB,5,2), REDUCE(MAXB,5,2), SAVE(MAXB,5,2),
       X  WHEN2, HPEO1, HPEO2, AVGTT, LANDNG(MAXB), AIDALT(MAXT),
       X  FILLER(MAXT,2), CKFILL(MAXT), PERIOD(20,3), REFILL(2,9),
       X  NOR(MAXB), NORHRS(MAXB),ALTPEO(NOPEOP,3),ALTAGE(NOAGE,3),
       X  CTPEOP(NOPEOP,5), AQPEOP(NOPEOP,5), CTPEO(NOPEOP),
       X  RECORD(NSCROL,3,MAXREC)
C
C
C   ************     AIRCRAFT, CREW AND FLIGHT RELATED ARRAYS   ************
C
C
C
        COMMON  / BASIC3 /ACN(MAXACN,40), ACDATA(30,MAXT), MXTASK(9),
       X   ACMDTA(20,MAXM,MAXT), BASES(50,MAXB),  WXDATA(WXDAYS,2,MAXB),
       X   PILOT(5,NOCREW), PILOTS(5,MAXT,MAXB), FLTRQT(LFQ,10),
       X   PRDFLT(MAXPER,5), SVEFLT(12,5), TEMPF( 50,4), FTAXI(2,MAXB),
       X   SCLP(5,MAXM,MAXT,2),  SCLRQT(NOSCL,9), TCONF(MAXM,MAXM,MAXT),
       X   SEEDED(10),
C
C   ************         QUEUE AND HEAP CONTROL VARIABLES         ************
C
N      X    LTQ,NET,FET,LET,OVERT,      LRQ,NER,FER,LER,OVERR,   OVERK,
N      X    LIQ,NEI,FEI,LEI,OVERI,      LWQ,NEW,FEW,LEW,OVERW,
N      X    LNT,NEN,FEN,LEN,OVERN,NOLD, LDT,NED,FED,LED,OVERD,DOLD,
N      X   LNOR,NEO,FEO,LEO,OVERO,      LFQ,NEF,FEF,LEF,OVERF, EF, LF,
N      X    LBQ,NEB,FEB,LEB,OVERB,      LLQ,NEL,FEL,LEL,OVERL,     NEH,
N      X    LGQ,NEG,FEG,LEG,OVERG,   NJOINT,NEJ,FEJ,LEJ,OVERJ,   OVERM,
N      X NLIMBO,NEX,FEX,LEX,OVERX,    NOCHG,NEV,FEV,LEV,OVERV,
C
C   ************           QUEUE AND HEAP STORAGE ARRAYS          ************
C
       X     TASKQ( LTQ,16),       REPQ( LRQ,11),        BUILDQ(LBQ,10),
       X     INTTSK( LIQ,10),      WAITSK( LWQ,13),      BACKLG(5, LLQ),
       X     NORQ(LNOR,3),         RQDTSK( LNT, 2),      DEFTSK( LDT,4),
       X     LIMBO(NLIMBO,6),      CANNTM( NOPART),      RESUPP( LGQ,5),
```

```
     X      REJOIN(NJOINT,2),      CHANGE(NOCHG,5)
C
C    *************       BLOCK DATA
C
N         COMMON / BASIC4 /  SEEDS, TSKCRT(99,5), RELIMP(33,5)
C
C    ******************************************************************************
C    ******************************************************************************
C
          COMMON / STOCKS /SHOPS(26,30,MAXB), PEOPLE(NOPEOP,7,MAXB),
     X    AGESTK(NOAGE,3,MAXB),PARTS(NOPART,5,MAXB),MUNSTK(NOMUN,4,MAXB),
     X    TRAP(NOTRAP,MAXB), MATERL(NOMATL,MAXB), POLSTK(MAXB),
     X    BORROW(NOUSER,2), DEPOT1(NOPEOP), DEPOT2(NOAGE),
     X    DEPOT3(NOPART), DEPOT4(NOMUN), DEPOT5(NOTRAP), DEPOT6(NOMATL),
     X    DEPOT7, DEPOT8(MAXT)
C
C    ******************************************************************************
C    ******************************************************************************
C
          COMMON / LOAD /  CONFIG(NOCONF,8), ACA(3,MAXM,MAXT,MAXB),
     X    ALERT(6,MAXM,MAXT,MAXB), PTZ(MAXM,MAXT,MAXB),
     X    SPARE(2,MAXM,MAXT,MAXB), SORDEF(16,3,MAXM,MAXT,MAXB)
C
C    ******************************************************************************
C    ******************************************************************************
C
N         INTEGER *4 NOTASK
C
N        COMMON /JOBS/ NOTASK, IGNORE, VBREAK, EXPED, ZSHOP,
     X   SHPORD(50,MAXT,MAXB), SHPTSK(3,NOTASK,25,MAXT),
     X   ZTASKS(5,MAXT,MAXB), TSKPR(25,MAXT,3), TSKRQT(NOTSK,15),
     X   TSKALT(NOTSKA,5), REPRQT(NOREP,8), REPALT(NOREPA,4),
     X   AGEREP(NOAGER,6), MUNRQT(4,NOBILD), LISTIN(LTHLST),
     X   PRTLST(NOPRT), ROOTS(NOPART,MAXT), CIRFTM(24),
     X   ADELAY(24,2,MAXB), AVGP(3,30,MAXB), JOBDTA(20,2)
C
C    ******************************************************************************
C    ******************************************************************************
C
N         COMMON / THEATR /  ORDER1,ORDER2, C4TM,C4INT, FD,LD, FA,LA,
N    X     TCONUS, NCONUS, NOCONS, NCARGO, NOSHP, NESHP, INDEX,
N    X     NXSCH,   NES, FES, LES,  OVERS,  NOSHIP,
N    X     NOPKG, PKGTM,   NEP, FEP, LEP,  OVERP,  OVERA,
     X     SHIPTO(MAXB,20,2), SHIPSC(NOSHP,3), SHIP(NOSHIP,7),
     X     SHIPQ(NOPKG,3), AVGSHP(MAXB), SHPT(MAXB,MAXB,3),
     X     SHIPTM(MAXB,MAXB,3), CONUS(NOCONS,2), CARGO(NCARGO,2)
C
C    ******************************************************************************
C    ******************************************************************************
C
          COMMON /BOMBSE/ FRACJB(NOFAC),
N    X                NOFAC,MXFAC,NOATT,LTHATT,NOITEM,
```

```
N     X    LTHCEQ,NOCE,NEC, FEC, LEC, OVERC, CEWORK, CEPEO, CEAGE, CRBLDG,
      X    ATRISK,CEDELY,SHPDLY,ATTDLY(2,MAXB),ATTACK(LTHATT,5),
      X    DAMAGE(NOITEM,2),CEPRTY(NOFAC),FACDAM(NOFAC,6),
      X    FACLTY(7,NOFAC,MAXB),CERQTS(8,NOCE), CEJOBQ(LTHCEQ,9)
C
C     ***************************************************************************
C     ***************************************************************************
C
N          INTEGER *4 NOREPT
N          COMMON / INFO / NOREPT, NORPT, OLDATA, NEWDTA,
      X          REPORT(NOREPT,4), XMIT(4,MAXB), PEORPT(2,NOPEOP,MAXB),
C
C
      X          AGERPT(NOAGE,MAXB), PRTRPT(4,NOPART,MAXB)
C
C     ***************************************************************************
C     ***************************************************************************
C
N          INTEGER *4  OUTSHP, OUTPER, OUTAGE, OUTPRT, OUTMUN,
N     X        OUTTRP, OUTMAT, OUTPOL, OUTFAC, OUTPT3, OUTPT4, OUTPT5
C
           COMMON/ OUT /     AVGTSK(25,MAXT), AVGREP(25,MAXT),
      X  XSTAT(10,30,MAXB),XXSTAT(10,30),
      XOUTPT4(2,30,MAXM,MAXB),OUTPER(2,NOPEOP,MAXB),OUTAGE(2,NOAGE,MAXB),
      X  OUTPRT(2,NOPART,MAXB), OUTMUN(2,NOMUN,MAXB),
      X  OUTTRP(2,NOTRAP,MAXB), OUTMAT(2,NOMATL,MAXB),
      X  OUTPOL(2,MAXB), OUTFAC(2,30,MAXB), OUTSHP(9,30,MAXB),
      X  OUTPT5(3,30,MAXB), OUTPT3(2,MAXM,MAXT,MAXB),
      X  OUTPT1(5,6,MAXM,MAXT,MAXB),
      X  OUTPT2(3,3,25,MAXB), CANCEL(5,MAXT,MAXB), SORTHR(24,MAXB),
      X    BSOR(MAXB), CUMSOR
C
C     ***************************************************************************
C     ***************************************************************************
C
N          INTEGER *4    PEORQT,  AGERQT
C
          COMMON/ REQTS /PEORQT(NOPEOP,MAXM,MAXT),AGERQT(NOAGE,MAXM,MAXT),
      X        SHOPRQ(30,MAXM,MAXT), PARTRQ(NOPART,MAXT),
      X        MUNRQD(NOMUN,MAXM,MAXT), TRAPRQ(2,3,MAXT),
      X        CANFLY(3,MAXM,MAXT,MAXB),
      X        SORCAP(MAXT,MAXB), TRYFLY( 6,MAXT,MAXB),
N     X        STATE, SELECT, MULTI1, MULTI2
C
C     ***************************************************************************
C     ***************************************************************************
C
           COMMON / CPARTS / BPARTS(15,MAXT,MAXB), POLICY(NOPART,MAXB,2),
N     X             OUTFIT,PMODE,PPRINT,ISHORT,NOPIPE,HIATUS,RANDM,CHNRTS
C
C
N          COMMON / AISCOM / NOSTAT, NOTRAY,
```

```
     X         AISDTA(NOSTAT,5), NSTAT(NOSTAT,2,MAXB),
     X         AISUSE(NOSTAT,6,MAXB),TRAY(NOPART),
     X         TRAYS(NOTRAY), TRAYST(NOTRAY,2,MAXB)
C
C    ******************************************************************************
C    ******************************************************************************
C
         COMMON / PURGE1 / COSTS(NOPART), PRTRQ(NOPART,2,MAXT),
     X   OFFBSE(2,50,2,MAXT),MAXOFF(2,MAXT), JOBPR(2,MAXT), ITEMS(MAXB),
     X   TPART(EXTPRT,3,MAXB)
C
C    ******************************************************************************
C    ******************************************************************************
C
         COMMON / PURGE2 / OFFMOB(NOPART,MAXT), OFFCOB(NOPART,MAXT),
     X     GTLMT(NOPART)
C
C    ******************************************************************************
C    ******************************************************************************
C
N        REAL   *4   IPIPE, INPIPE, FRACBS, CSTOCK, SHORT
C
         COMMON / PURGE3 / TOTALS(NOPART,MAXB,3), TOCIRF(NOPART,2),
     X    IPIPE(NOPART,2), INPIPE(NOPART,MAXB,2), FRACBS(NOPART,MAXB),
     X    CSTOCK(NOPART,2),SHORT(NOPART),SRFRAC(MAXB,2),CHCKED(NOPART)
C
C    ******************************************************************************
C    ******************************************************************************
C
C        THE PURGE4 COMMON IS ONLY USED LOCALLY IN SUBROUTINES BOMB
C        AND REORGN BUT IS TREATED AS COMMON SO THAT IT WILL BE
C        REDIMENSIONED AS REQUIRED.  IT NEED NOT BE ZEROED NOR STORED.
C
         COMMON / PURGE4 / STAFF(NOPEOP,2), SICK(NOPEOP),SHOPEO(NOPEOP),
     X                     SHOPAG(NOAGE)
C
C    ******************************************************************************
C    ******************************************************************************
C
C        THE PURGE5 COMMON IS ONLY USED LOCALLY IN SUBROUTINE ZSHOPS
C        BUT IS TREATED AS COMMON SO THAT IT WILL BE REDIMENSIONED
C        AS REQUIRED.  IT NEED NOT BE ZEROED NOR STORED.
C
N        INTEGER *4    ZPRTRQ
C
         COMMON / PURGE5 / ZPRTRQ(NOPART)
C
C    ******************************************************************************
C    ******************************************************************************
C
       COMMON/TESTS/TTRIAL,TEST1,START(6),STOP(6)
C
```

```
C
C
C
N      EQUIVALENCE (Z1(1),MAXACN), (Z2(1),ACN(1,1)), (Z3(1),SEEDS(1)),
N     X (Z4(1),SHOPS(1,1,1)),   (Z5(1),CONFIG(1,1)),   (Z6(1),NOTASK),
N     X (Z7(1),ORDER1),     (Z8(1),FRACJB(1)),     (Z9(1),NOREPT),
N     X (Z10(1),AVGTSK(1,1)), (Z11(1),PEORQT(1,1,1)),
N     X (Z12(1),BPARTS(1,1,1)), (Z13(1),NOSTAT), (Z14(1),TTRIAL),
N     X (Z15(1),COSTS(1)), (Z16(1),OFFMOB(1,1)), (Z17(1),TOTALS(1,1,1))
C
C
```

```
**********************************************
The exemplary Common listed below was
developed for use on a CDC CYBAR 176.
**********************************************
```

```
C
C
C    ********************************************************************************
C    ********************************************************************************
*COMDECK BASIC
C
N      REAL    *4        SEEDS(10)
N      INTEGER *4        MAXACN,MAXB,MAXT,MAXM,NOTSK,NOTSKA,NOREP,NOREPA,
N     X      NOAGER,NOBILD,NOPEOP,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOSCL,
N     X      NOCONF,NOPRT,LTHLST,NXSEED,RINDEX(4,9),
      X      LATERL(MAXB),PRTCRT(NOPRT,2)
C
C    ** COMMONS BASIC1, BASIC2, BASIC3 AND BASIC4 ARE INCLUDED BY "BASIC"
C
C
N      COMMON  /BASIC1/ MAXACN,MAXB,MAXT,MAXM,NOTSK,NOTSKA,NOREP,NOREPA,
N     X  NOAGER,NOBILD,NOPEOP,NOAGE,NOPART,NOMUN,NOTRAP,NOMATL,NOSCL,
N     X  NOCONF,NOPRT,LTHLST, NXSEED, RINDEX, LATERL
C
C    ***********            KEY CONTROL VARIABLES                ***********
C
N        COMMON / BASIC2 /   NOW, HOUR, NBASE, NTYPE, NPART, CREWS,
N     X  SIMLTH, EXTEND, NTRIAL, ITRIAL, PRINT, STATFQ, CUMSTA, OVERFL,
N     X  TSAR, FIXAGE, BUILD, AIDA, DAMODE, NOSAVE, FSALVG,
N     X  TEST, XTEST, SEED, VERIFY, IDAY, NOUSER, NONUNI, DOWNTM, CANSRU,
N     X  SCROLL, NSCROL, SCROL1, SCROL2, TOOFEW, K1LOW, K2LOW,
N     X  SLEEP, REST, NOCREW, NPILOT, MAXFLT, MAXPER, MULTI, CANFLT,
N     X  NOPOMO,ORDIT,ORDWT,CMODE,NOFUEL,CCIRF,CTHEA,SHOPRY,CIRFLG,
N     X  UNCER, CONSIG, TODOCK, ADAPTR, SHPREP, SEEKSH, NRTPOL, PROTME,
N     X  OPSBSE, EMERG, POSTPN, CONCUR, LTHDEF, ALERTR, DELYPF, OVERTM,
N     X  JOBCON, FILLAC, FLEVEL, MNTLMT, MNTF, MNTR, QUIK, RPARTS,MAXMNT,
N     X  NEWPRT, ZNORS, LOSTAC, MAXTME,
N     X  CANMOD, CANMUL, CANCAN, DOCANN, MXHOLE, CDELAY,
N     X  MAXREC,WXDAYS,TPLAN,NEXTIN,NEXTSC,ENDAY,LOADTM,LSTTOD,DMDLMT,
```

```
N     X  SPARE1,SPARE2,SPARE3,SPARE4,SPARE5,SPARE6,SPARE7,SPARE8,SPARE9,
      X  HURRY(MAXB,5,2), REDUCE(MAXB,5,2), SAVE(MAXB,5,2),
      X  WHEN2, HPEO1, HPEO2, AVGTT, LANDNG(MAXB), AIDALT(MAXT),
      X  FILLER(MAXT,2), CKFILL(MAXT), PERIOD(20,3), REFILL(2,9),
      X  NOR(MAXB), NORHRS(MAXB),ALTPEO(NOPEOP,3),ALTAGE(NOAGE,3),
      X   RECORD(NSCROL,3,MAXREC)
C
C
C     ***********      AIRCRAFT, CREW AND FLIGHT RELATED ARRAYS   ***********
C
C
C
          COMMON  / BASIC3 /   ACN(MAXACN,40), ACDATA(30,MAXT),
      X   MXTASK(9), ACMDTA(20,MAXM,MAXT), BASES(50,MAXB),
      X   PILOT(5,NOCREW), PILOTS(5,MAXT,MAXB), FLTRQT(LFQ,10),
      X   SVEFLT(12,5),    TEMPF( 50,4),  FTAXI(2,MAXB),
      X   SCLP(5,MAXM,MAXT,2),  SCLRQT(NOSCL,9), TCONF(MAXM,MAXM,MAXT),
      X   SEEDED(10),
C
C     ***********         QUEUE AND HEAP CONTROL VARIABLES         ***********
C
N     X    LTQ,NET,FET,LET,OVERT,       LRQ,NER,FER,LER,OVERR,   OVERK,
N     X    LIQ,NEI,FEI,LEI,OVERI,       LWQ,NEW,FEW,LEW,OVERW,
N     X    LNT,NEN,FEN,LEN,OVERN,NOLD, LDT,NED,FED,LED,OVERD,DOLD,
N     X   LNOR,NEO,FEO,LEO,OVERO,       LFQ,NEF,FEF,LEF,OVERF, EF, LF,
N     X    LBQ,NEB,FEB,LEB,OVERB,       LLQ,NEL,FEL,LEL,OVERL,     NEH,
N     X    LGQ,NEG,FEG,LEG,OVERG,    NJOINT,NEJ,FEJ,LEJ,OVERJ,   OVERM,
N     X NLIMBO,NEX,FEX,LEX,OVERX,     NOCHG,NEV,FEV,LEV,OVERV,
C
C     ***********          QUEUE AND HEAP STORAGE ARRAYS          ***********
C
      X     TASKQ( LTQ,16),        REPQ( LRQ,11),        BUILDQ(LBQ,10),
      X     INTTSK( LIQ,10),       WAITSK( LWQ,13),      BACKLG(5, LLQ),
      X     NORQ(LNOR,3),          RQDTSK( LNT, 2),      DEFTSK( LDT,4),
      X     LIMBO(NLIMBO,6),       CANNTM( NOPART),      RESUPP( LGQ,5),
      X     REJOIN(NJOINT,2),      CHANGE(NOCHG,5)
C
C     ************      BLOCK DATA
C
N         COMMON / BASIC4 /  SEEDS, TSKCRT(99,5), RELIMP(33,5)
C
N         LEVEL 2,LBASIC
C
          COMMON / LBASIC / PRTCRT( NOPRT,2), CTPEOP(NOPEOP,5),
      X   AQPEOP(NOPEOP,5), CTPEO(NOPEOP), WXDATA(WXDAYS,2,MAXB),
      X   PRDFLT(MAXPER,5)
C
C     ************************************************************************
C     ************************************************************************
```

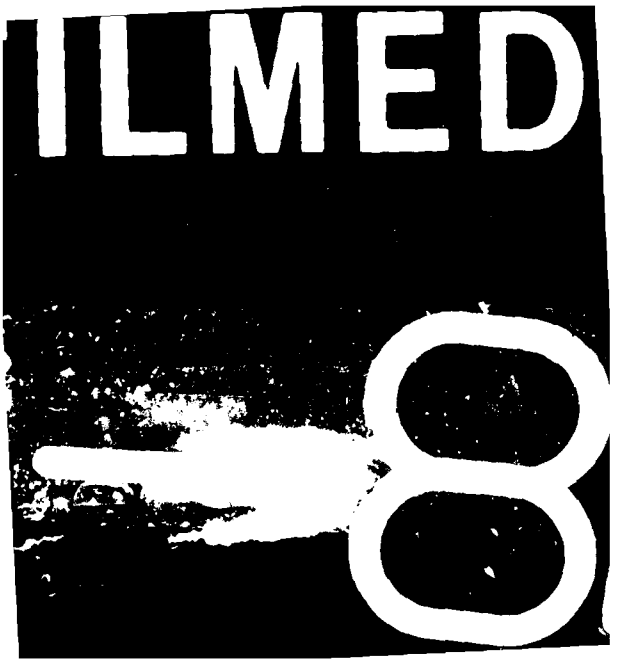
```
*COMDECK STOCKS
C
C
          COMMON / STOCKS /SHOPS(26,30,MAXB), PEOPLE(NOPEOP,7,MAXB),
     X    AGESTK(NOAGE,3,MAXB),PARTS(NOPART,5,MAXB),MUNSTK(NOMUN,4,MAXB),
     X    TRAP(NOTRAP,MAXB), MATERL(NOMATL,MAXB), POLSTK(MAXB),
     X    BORROW(NOUSER,2),
     X    DEPOT7, DEPOT8(MAXT)
C
N         LEVEL 2,LSTOCK
C
          COMMON / LSTOCK / DEPOT1(NOPEOP), DEPOT2(NOAGE),
     X    DEPOT3(NOPART), DEPOT4(NOMUN), DEPOT5(NOTRAP), DEPOT6(NOMATL)
C
C    ******************************************************************************
C    ******************************************************************************
*COMDECK LOAD
C
          COMMON / LOAD /  CONFIG(NOCONF,8), ACA(3,MAXM,MAXT,MAXB),
     X    ALERT(6,MAXM,MAXT,MAXB), PTZ(MAXM,MAXT,MAXB),
     X    SPARE(2,MAXM,MAXT,MAXB), SORDEF(16,3,MAXM,MAXT,MAXB)
C
C    ******************************************************************************
C    ******************************************************************************
*COMDECK JOBS
C
N         INTEGER *4 NOTASK
C
N         COMMON /JOBS/ NOTASK, IGNORE, VBREAK, EXPED, ZSHOP,
     X   SHPORD(50,MAXT,MAXB), SHPTSK(3,NOTASK,25,MAXT),
     X   TSKPR(25,MAXT,3), TSKRQT(NOTSK,15),
     X    REPRQT(NOREP,8),
     X   AGEREP(NOAGER,6), MUNRQT(4,NOBILD),
     X   ADELAY(24,2,MAXB), AVGP(3,30,MAXB), JOBDTA(20,2)
C
N         LEVEL 2,LJOBS
C
          COMMON / LJOBS / ZTASKS(5,MAXT,MAXB), TSKALT(NOTSKA,5),
     X       REPALT(NOREPA,4), LISTIN(LTHLST),
     X   PRTLST(NOPRT), ROOTS(NOPART,MAXT), CIRFTM(24)
C
C    ******************************************************************************
C    ******************************************************************************
*COMDECK THEATR
C
N         COMMON / THEATR /  ORDER1,ORDER2, C4TM,C41NT, FD,LD, FA,LA,
N     X    TCONUS, NCONUS, NOCONS, NCARGO, NOSHP, NESHP, INDEX,
N     X    NXSCH,   NES, FES, LES,  OVERS,  NOSHIP,
N     X    NOPKG, PKGTM,   NEP, FEP, LEP,  OVERP,  OVERA,
      X    SHIP(NOSHIP,7)
C
N         LEVEL 2,LTHEAT
C
          COMMON / LTHEAT / SHIPTO(MAXB,20,2), SHIPSC(NOSHP,3),
     X    SHIPQ(NOPKG,3), AVGSHP(MAXB), SHPT(MAXB,MAXB,3),
```

```
     X     SHIPTM(MAXB,MAXB,3), CONUS(NOCONS,2), CARGO(NCARGO,2)
C
C    ******************************************************************************
C    ******************************************************************************
*COMDECK BOMBSE
C
N         COMMON /BOMBSE/ NOFAC, MXFAC, NOATT,LTHATT,NOITEM,LTHCEQ,NOCE,
N    X    NEC, FEC, LEC, OVERC, CEWORK, CEPEO, CEAGE, CRBLDG, ATRISK,
     X    CEDELY, SHPDLY, ATTDLY(2,MAXB),ATTACK(LTHATT,5),
     X    CEJOBQ(LTHCEQ,9)
C
N         LEVEL 2,LBOMB
C
          COMMON / LBOMB /  FRACJB(NOFAC),DAMAGE(NOITEM,2),CEPRTY(NOFAC),
     X    FACDAM(NOFAC,6), FACLTY(7,NOFAC,MAXB), CERQTS(8,NOCE)
C
C    ******************************************************************************
C    ******************************************************************************
*COMDECK INFO
C
N         INTEGER *4 NOREPT
C
N         COMMON / INFO / NOREPT,NORPT,OLDATA,NEWDTA,
     X         REPORT(NOREPT,4)
C
N         LEVEL 2,LINFO
C
          COMMON / LINFO /  XMIT(4,MAXB),   PEORPT(2,NOPEOP,MAXB),
     X         AGERPT(NOAGE,MAXB), PRTRPT(4,NOPART,MAXB)
C
C    ******************************************************************************
C    ******************************************************************************
*COMDECK OUT
C
N         INTEGER *4  OUTSHP, OUTPER, OUTAGE, OUTPRT, OUTMUN,
N    X        OUTTRP, OUTMAT, OUTPOL, OUTFAC, OUTPT3, OUTPT4, OUTPT5
C
          COMMON / OUT /    AVGTSK(25,MAXT), AVGREP(25,MAXT),
     X       CANCEL(5,MAXT,MAXB), SORTHR(24,MAXB), BSOR(MAXB), CUMSOR
C
N         LEVEL 2,LOUT
C
          COMMON / LOUT /  OUTPT4( 2,30,MAXM,MAXB),
     X       XSTAT(10,30,MAXB),XXSTAT(10,30),
     X       OUTPER(2,NOPEOP,MAXB), OUTAGE(2,NOAGE,MAXB),
     X       OUTPRT(2,NOPART,MAXB), OUTMUN(2,NOMUN,MAXB),
     X       OUTTRP(2,NOTRAP,MAXB), OUTMAT(2,NOMATL,MAXB),
     X       OUTPOL(2,MAXB), OUTFAC(2,30,MAXB), OUTSHP(9,30,MAXB),
     X       OUTPT5(3,30,MAXB), OUTPT3(2,MAXM,MAXT,MAXB),
     X       OUTPT1(5,6,MAXM,MAXT,MAXB),
     X       OUTPT2(3,3,25,MAXB)
C
```

```
C     ******************************************************************************
C     ******************************************************************************
*COMDECK REQTS
C
N          INTEGER *4    PEORQT,  AGERQT
C
N          LEVEL 2,REQTS
C
          COMMON/ REQTS /PEORQT(NOPEOP,MAXM,MAXT),AGERQT(NOAGE,MAXM,MAXT),
     X         SHOPRQ(30,MAXM,MAXT), PARTRQ(NOPART,MAXT),
     X         MUNRQD(NOMUN,MAXM,MAXT), TRAPRQ(2,3,MAXT),
     X         CANFLY(3,MAXM,MAXT,MAXB),
     X         SORCAP(MAXT,MAXB), TRYFLY( 6,MAXT,MAXB),
N    X         STATE, SELECT, MULTI1, MULTI2
C
C     ******************************************************************************
C     ******************************************************************************
*COMDECK CPARTS
C
N          LEVEL 2,CPARTS
C
          COMMON / CPARTS / BPARTS(15,MAXT,MAXB),OUTFIT,
N    X               PMODE,PPRINT,ISHORT,NOPIPE,HIATUS,RANDM,CHNRTS
C
C     ******************************************************************************
C     ******************************************************************************
*COMDECK AISCOM
C
N          LEVEL 2,AISCOM
C
N          COMMON / AISCOM / NOSTAT, NOTRAY,
     X          AISDTA(NOSTAT,5), NSTAT(NOSTAT,2,MAXB),
     X          AISUSE(NOSTAT,6,MAXB),TRAY(NOPART),
     X          TRAYS(NOTRAY), TRAYST(NOTRAY,2,MAXB)
C
C     ******************************************************************************
C     ******************************************************************************
*COMDECK PURGE1
C
N          LEVEL 2,PURGE1
C
          COMMON / PURGE1 / COSTS(NOPART), PRTRQ(NOPART,2,MAXT),
     X    OFFBSE(2,50,2,MAXT),MAXOFF(2,MAXT), JOBPR(2,MAXT), ITEMS(MAXB),
     X    TPART(EXTPRT,3,MAXB)
C
C     ******************************************************************************
C     ******************************************************************************
*COMDECK PURGE2
C
N          LEVEL 2,PURGE2
C
          COMMON / PURGE2 / OFFMOB(NOPART,MAXT), OFFCOB(NOPART,MAXT),
     X      GTLMT(NOPART)
```

```
C
C     ******************************************************************************
C     ******************************************************************************
*COMDECK PURGE3
C
N          LEVEL 2,PURGE3
C
N          REAL   *4    IPIPE, INPIPE, FRACBS, CSTOCK, SHORT
C
           COMMON / PURGE3 / TOTALS(NOPART,MAXB,3), TOCIRF(NOPART,2),
     X      IPIPE(NOPART,2), INPIPE(NOPART,MAXB,2), FRACBS(NOPART,MAXB),
     X      CSTOCK(NOPART,2), SHORT(NOPART), POLICY(NOPART,MAXB,2),
     X      SRFRAC(MAXB,2), CHCKED(NOPART)
C
C     ******************************************************************************
C     ******************************************************************************
*COMDECK PURGE4
C
N          LEVEL 2,PURGE4
C
C          THE PURGE4 COMMON IS ONLY USED LOCALLY IN SUBROUTINES BOMB
C          AND REORGN BUT IS TREATED AS COMMON SO THAT IT WILL BE
C          REDIMENSIONED AS REQUIRED.  IT NEED NOT BE ZEROED NOR STORED.
C
           COMMON / PURGE4 / STAFF(NOPEOP,2), SICK(NOPEOP),SHOPEO(NOPEOP),
     X     SHOPAG(NOAGE)
C
C     ******************************************************************************
C     ******************************************************************************
*COMDECK PURGE5
C
N          LEVEL 2,PURGE5
C
C          THE PURGE5 COMMON IS ONLY USED LOCALLY IN SUBROUTINE ZSHOPS
C          BUT IS TREATED AS COMMON SO THAT IT WILL BE REDIMENSIONED
C          AS REQUIRED.  IT NEED NOT BE ZEROED NOR STORED.
C
N          INTEGER *4     ZPRTRQ
C
           COMMON / PURGE5 / ZPRTRQ(NOPART)
C
C     ******************************************************************************
C     ******************************************************************************
C
        COMMON/TESTS/TTRIAL,TEST1,START(6),STOP(6)
C
C
C
N      EQUIVALENCE (Z1(1),MAXACN), (Z2(1),ACN(1,1)), (Z3(1),SEEDS(1)),
N     X (Z4(1),SHOPS(1,1,1)),    (Z5(1),CONFIG(1,1)),    (Z6(1),NOTASK),
N     X (Z7(1),ORDER1),      (Z8(1),FRACJB(1)),      (Z9(1),NOREPT),
```

```
N     X (Z10(1),AVGTSK(1,1)), (Z11(1),PEORQT(1,1,1)),
N     X (Z12(1),BPARTS(1,1,1)), (Z13(1),NOSTAT), (Z14(1),TTRIAL),
N     X (Z15(1),COSTS(1)), (Z16(1),OFFMOB(1,1)), (Z17(1),TOTALS(1,1,1))
C
N      EQUIVALENCE (L1(1),PRTCRT(1,1)), (L2(1),DEPOT1(1)),
N     X     (L3(1),ZTASKS(1,1,1), (L4(1),SHIPTO(1,1,1),
N     X     (L5(1), DAMAGE(1,1)), (L6(1), XMIT(1,1)),)),
N     X     (L7(1), OUTPT4(1,1,1,1))
C
```


ILMED